REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
09-05-2012		

4. TITLE AND SUBTITLE
Measurement of Ship Air Wake Impact on a Remotely Piloted Vehicle

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Metzger, Jason Daniel

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Trident Scholar Report no. 406 (2012)

12. DISTRIBUTION/AVAILABILITY STATEMENT
This document has been approved for public release; its distribution is UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

For this Trident Project, a remotely piloted helicopter has been used to investigate the turbulent air wake away from the flight deck behind a ship. As the helicopter maneuvers through regions in the ship's air wake where there are steep velocity gradients, an inertial measurement unit mounted on the helicopter records a noticeable change in the helicopter's flight path. At any time, the relative position of the helicopter can be determined by comparing GPS derived position of the helicopter with that of a reference position on the ship. Combining these two measurement systems, the locations of sharp gradients in the air wake can be mapped relative to the ship and compared with CFD simulations of similar wind-over-deck configurations.

15. SUBJECT TERMS
helicopter, ship air wake, CFD, computational fluid dynamics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		66	19b. TELEPHONE NUMBER *(Include area code)*

A TRIDENT SCHOLAR PROJECT REPORT

NO. 406

Measurement of Ship Air Wake Impact on a Remotely Piloted Aerial Vehicle

by

Midshipman 1/C Jason D. Metzger, USN

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

USNA-1531-2

U.S.N.A. --- Trident Scholar project report; no. 406 (2012)

MEASUREMENT OF SHIP AIR WAKE IMPACT ON A REMOTELY PILOTED AERIAL VEHICLE

by

Midshipman 1/c Jason D. Metzger
United States Naval Academy
Annapolis, Maryland

_____
(signature)

Certification of Advisers Approval

Captain Murray R. Snyder, USN
Permanent Military Professor, Mechanical Engineering Department

_____
(signature)

_____
(date)

Visiting Professor John S. Burks
Office of Naval Research Rotorcraft Chair, Aerospace Engineering Department

_____
(signature)

_____
(date)

Dr. Hyung S. Kang
Assistant Research Professor, Aerospace Engineering Department

_____
(signature)

_____
(date)

Acceptance for the Trident Scholar Committee

Professor Carl E. Wick
Associate Director of Midshipman Research

_____
(signature)

_____
(date)

USNA-1531-2

Abstract

This Trident Scholar project complemented an ongoing research program being conducted at the United States Naval Academy (USNA) involving an investigation of ship air wakes using an instrumented training and patrol (YP) craft. The objective of the program is to validate Computational Fluid Dynamics (CFD) tools that will be useful in determining ship air wake impact on naval rotary wing vehicles. Because the YPs are relatively large vessels with a similar superstructure and deck configuration to that of a cruiser or a destroyer, air wake data can be collected that corresponds well with that of modern naval warships. Data collected from both at-sea measurements and wind tunnel testing are being compared with CFD models already used to help predict ship air wake effects.

For this Trident project, a remotely piloted helicopter has been used to investigate the turbulent air wake away from the flight deck behind a ship. As the helicopter, which has a 4.5 ft diameter rotor, maneuvers through regions in the ship's air wake where there are steep velocity gradients, an inertial measurement unit mounted on the helicopter records a noticeable change in the helicopter's flight path. At any time, the relative position of the helicopter can be determined by comparing the GPS derived position of the helicopter with that of a reference position on the ship. Combining these two measurement systems, the locations of sharp gradients in the air wake can be mapped relative to the ship (accurate within one rotor diameter of the helicopter) and compared with CFD simulations of similar wind-over-deck configurations. Data collected from underway flight operations show a macro-scale validation of CFD predictions of the ship's air wake at locations distant from the flight deck for winds 15° and 30° off the starboard bow.

Keywords: helicopter ship air wake CFD interaction

Acknowledgments

This research is supported by the Office of Naval Research via the U.S. Naval Academy Trident Scholar Program. Program Officer is Prof. Carl Wick. The author would also like to acknowledge and thank the continued support and guidance of his advisors for the project. The Trident Scholar Committee is also thanked for their guidance and review throughout the project. A special thank you is also extended to the Naval Academy Fluids Lab Staff, specifically Ms. Louise Becnel, Mr. George Burton, Mr. Russel Foard, and Mr. Fritz Woolford. Ms. Cindi Gallagher of the Multimedia Support Center in Nimitz Library is also acknowledged for her assistance in preparing graphics and visualizations for this project. Also, the hard work and cooperation from the craftmaster and crew of YP676 was appreciated throughout the project. Additionally, the assistance received from Mr. Charles Rosario in his understanding and expertise of remote controlled helicopters was invaluable and necessary for the success of this project.

Table of Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

CCWT .....Closed Circuit Wind Tunnel
CFD.....Computational Fluid Dynamics
GPS .....Global Positioning System
IMU.....Inertial Measurement Unit
ONR .....Office of Naval Research
RC .....Remote Controlled
USNA.....United States Naval Academy
YP .....Patrol Craft, Training
β .....Relative wind over deck expressed as an azimuth (0° down the bow)

1 Background

The launch and recovery of rotary wing aircraft from naval vessels is a critical part of modern naval aviation. In the United States Navy, helicopters are used onboard ships ranging from nuclear-powered aircraft carriers to amphibious transport docks to frigates, destroyers, and cruisers, carrying out a variety of missions, including search and rescue, mine countermeasures, and tactical insertions. Considering that helicopters are launched from a flight deck on the aft end of a ship, which pitches and rolls as the ship transits through open seas, the actual launch and recovery of rotary wing aircraft can be challenging and potentially dangerous. Coupling the ship's movement with the turbulent air wake created by the ship's superstructure and the rotor wake produced by the helicopter presents a problem specific to each individual ship and aircraft combination.

To ensure safety for the aircraft, aircrew, ship's crew, and the ship itself, launch and recovery envelopes are developed for each specific aircraft and ship combination.[1,2] Figure 1 shows an example of a launch and recovery envelope in use today by MH-60S pilots landing onboard USS Ticonderoga (CG 47) class cruisers. The development of these envelopes requires expensive and potentially hazardous testing by naval test pilots, resulting in a diagram such as Figure 1, which details safe conditions for helicopter operations at various wind over deck speeds and bearings during both daytime and night operations. During testing, the pilots will make repeated approaches while the ship is maneuvering to provide a specific wind over deck speed and direction. The pilots will then subjectively assign scores to each approach, quantifying wind conditions the average fleet pilot should be capable of safely flying through. Because of the nature of the testing, the aircraft and aircrew are put through a potentially risky build-up approach in order to expand the launch and recovery envelopes to the maximum safe extent possible.

The time required for flight testing could be reduced and the risks involved could be mitigated through the use of computational tools to predict test conditions and extrapolate test results, thus reducing the number of test points and approaches flown by test pilots.[3] However, current computational methods have not been validated for ships with a complex superstructure, such as destroyers and cruisers.[4-10]

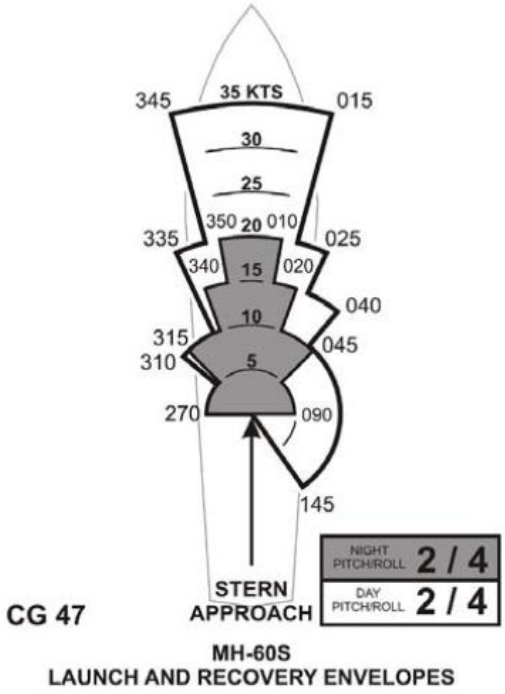

Figure 1. Launch and recovery envelopes for MH-60S helicopters on USS Ticonderoga (CG 47) class cruisers.[2]

1.1 Ship Air Wake Project

The Ship Air Wake Project,[3,11-13] an ongoing research study at the United States Naval Academy (USNA), funded by the Office of Naval Research (ONR) with coordination with Naval Air Systems Command, seeks to develop validated Computational Fluid Dynamics (CFD) tools that would reduce the amount of at-sea flight testing required, making rotary wing launch and recovery envelope development safer, more efficient, and more affordable. For a CFD analysis to be credible, the data collected from the analysis must be validated through additional experimental and situational measurements. The Ship Air Wake Project takes advantage of unique resources available at USNA that allow for a systematic analysis of ship air wakes.

1.1.1 In Situ Measurements of Ship Air Wakes

USNA operates a fleet of patrol craft (YP) for midshipmen training purposes. The YPs (length of 32.9 m [108 ft] and height above the waterline of 7.3 m [24 ft]), as seen in Figure 2, are vessels with a similar superstructure and deck layout to that of a modern destroyer or cruiser.

Because of the size of the YPs, air wake data can be collected with Reynolds numbers in the same order of magnitude as those for modern naval warships (Reynolds number is the ratio of inertia forces to viscous forces). As shown in Figure 3, YP676 has been modified to include a flight deck and hangar structure representative of what is currently found on naval vessels. Figure 4 contains the dimensions and a detailed layout of the modified flight deck and hangar structure.

Figure 2. Unmodified USNA YP.

Figure 3. YP676 with added flight deck.

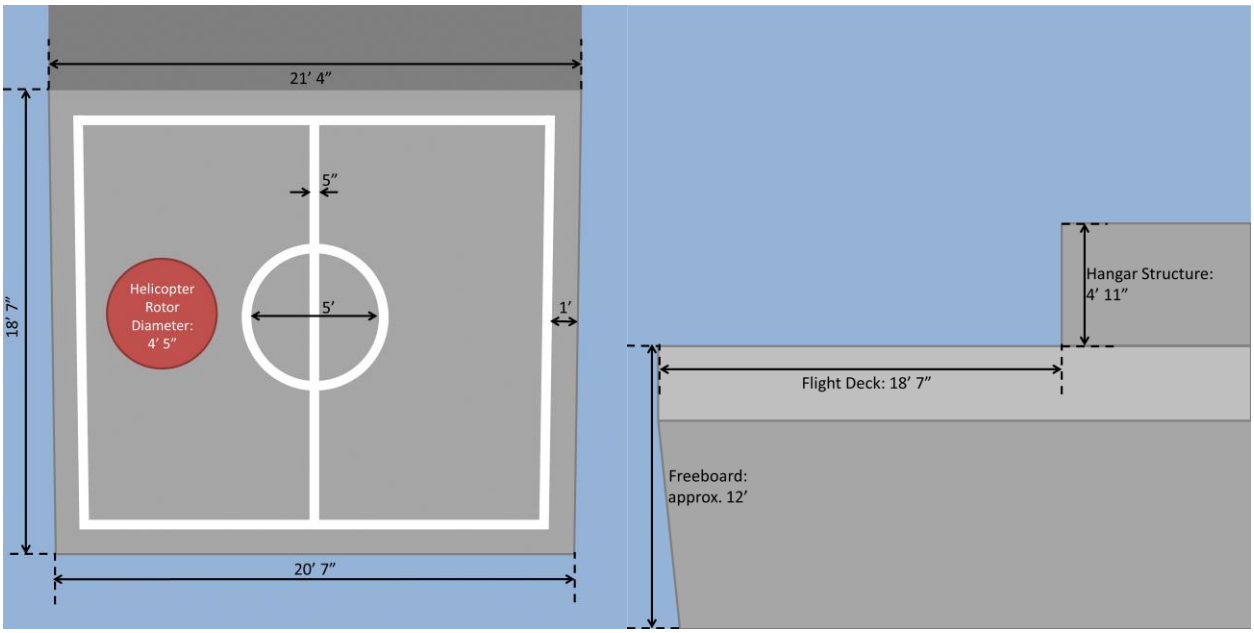

Figure 4. Detailed modified YP flight deck dimensions (left: top view; right: starboard side view). Red disc represents helicopter rotor diameter used for testing.

Ultrasonic anemometers have been installed to allow for direct measurement of wind speed and direction over the flight deck (Figure 5). The anemometers are accurate to ± 1.18 in/sec and are connected to a data logger unit that allows up to eight different anemometers to be sampled concurrently. To estimate the reference wind, or the wind unaffected by airflow over the ship, one anemometer is mounted 3.5 feet forward and 7.0 feet above the ship's bow (Figure 6). The reference wind is collected to allow comparison with similar CFD data collected at a given crosswind component (e.g., wind 15° off the starboard bow). Additional data collected include ship pitch and roll, ambient temperature, and atmospheric pressure. Through August 2011, 22 underway test periods have been completed in the Chesapeake Bay, providing flight deck data for comparison with CFD and wind tunnel data.

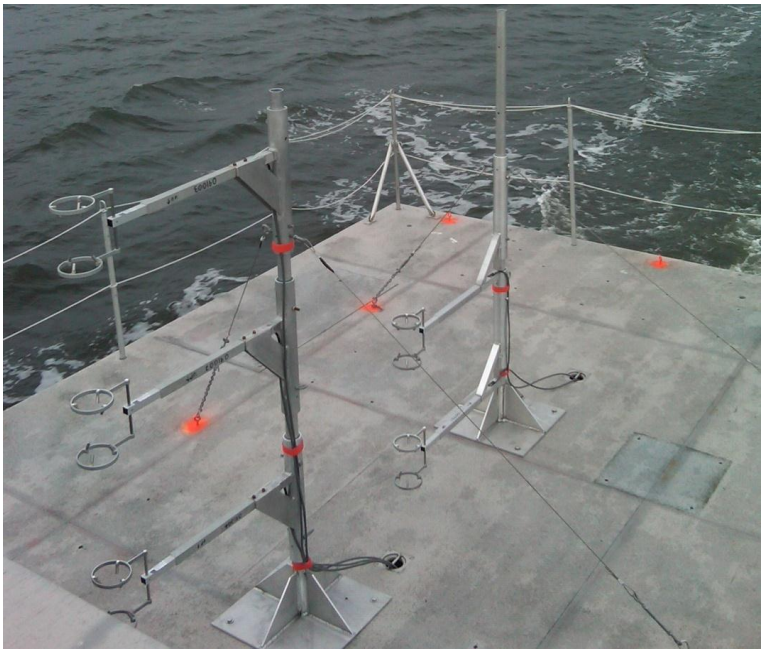

Figure 5. Ultrasonic anemometers installed on YP676 flight deck.

Figure 6. Bow reference wind anemometer.

1.1.2 Wind Tunnel Measurements

A 4% scale model of YP676 (Figure 7) has been crafted, allowing for thorough testing in USNA's Closed-Circuit Wind Tunnel (CCWT). Initial wind tunnel tests were conducted at a test

section velocity of 300 ft/sec, matching the Reynolds number encountered by the actual YP676 with a seven knot wind over deck.

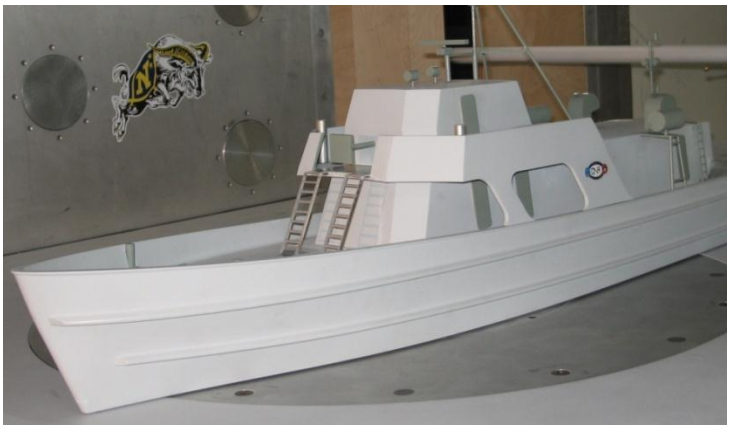

Figure 7. 4% scale model of YP676 installed in the USNA CCWT.

1.1.3 Computational Fluid Dynamics Simulations

Numerical simulations for various wind over deck parameters have been performed by USNA midshipmen with parallel processing using Cobalt, a commercial CFD code, and with Kestrel, a government CFD code. Both codes use an unstructured grid, as shown in Figure 8, which allows for finer resolution where greater variation in airflow is expected. This YP model, containing approximately fifteen million tetrahedrons, is used to complete the CFD analyses for the Ship Air Wake Project.

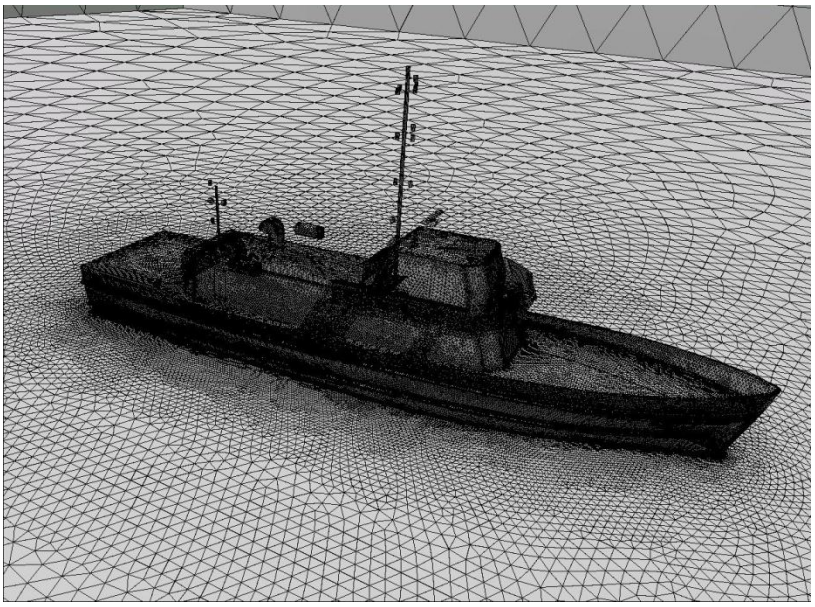

Figure 8. Unstructured YP surface grid.

1.1.4 Interim Results

Prior to the start of the Trident research project, 22 underway test and data collection periods had been completed.[11,13] Additionally, midshipmen had performed CFD analysis for 7, 12, and 20 knots of wind over deck for a headwind and for crosswinds from the starboard bow at various angles. A good comparison has been observed between normalized *in situ* and CFD simulation data for numerous locations above the flight deck; however, the comparisons between velocity direction is better than the velocity magnitude. These comparisons can be seen in Figure 9, which represents the CFD flow simulation by varying colors in the background in addition to white directional vectors in contrast to black vectors representing data measured underway. Good measurement repeatability has been observed over the 22 separate underway test periods.

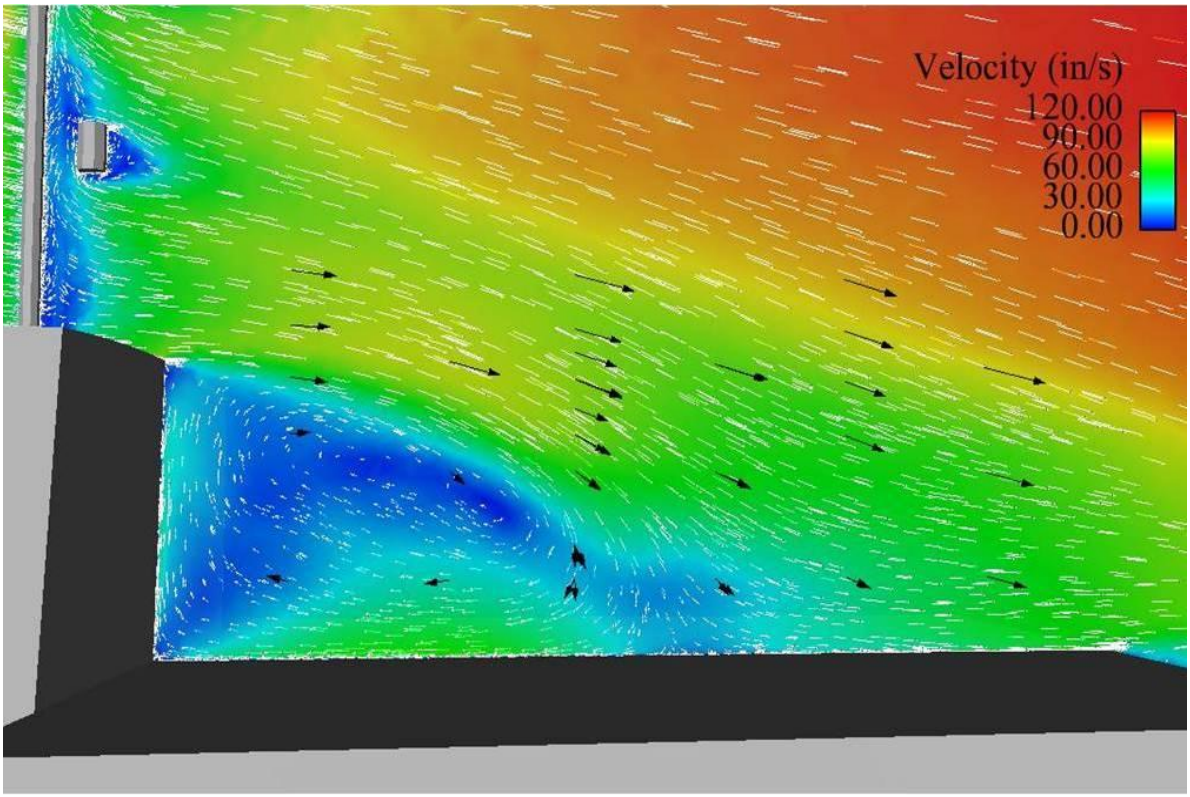

Figure 9. Centerline normalized *in situ* data (black vectors) vs. time-averaged CFD data (white vectors and color scale) for a 7-knot headwind.[11,13]

Since wind velocity data can only be collected in a few locations at a time, additional underway sessions are planned to cover additional areas of interest above the flight deck. However, collecting data away from the flight deck cannot be sufficiently accomplished through this method of investigation. There is a physical limit of about 15 feet to which the anemometers

can be extended away from the ship. According to CFD analyses, significant velocity changes on a macro scale are predicted at much further distances away from the flight deck. In order to validate the CFD analyses of a YP's ship air wake, velocity changes, which correspond to air wake turbulence, must be collected distant from the ship for comparison with computational predictions.

2 Method of Investigation

The proposed goal of this Trident project was to measure the impact of a YP's air wake on a remotely piloted rotorcraft and to compare the data with results obtained through computational simulations. Data about air velocities at specific locations in the air wake was not measured like on the flight deck; instead, through the integration of three separate systems, the path of a remotely piloted helicopter can be mapped where its flight is disturbed by the air wake. As the vehicle flies through areas of significant change in wind velocity, the rotorcraft exhibits disturbances in its flight characteristics, resulting in changes in aircraft orientation and airspeed. To acquire data on the helicopter's reaction to changes in air wake velocity, the rotorcraft was equipped with an Inertial Measurement Unit (IMU) that records gyroscopic rates and axial accelerations. Global Positioning System (GPS) units were placed on both the helicopter and the YP, allowing for the calculation of the relative position of the helicopter with respect to the YP at any given time. Combining these systems, the method of investigation employed allows for the mapping of locations relative to the YP where the helicopter was in the ship's air wake.

2.1 Equipment

Three primary sets of equipment and their associated accessories were used to conduct this research project. Three separate T-Rex 600 E Super Pro Remote Controlled (RC) helicopters were used throughout the duration of the project. Mounted to the helicopter during each flight was an x-IMU Inertial Measurement Unit, which recorded disturbances in the helicopter's flight. Also mounted on the helicopter was a SP GPS Data Logger, which, when paired with an identical GPS unit onboard the YP, allowed for the calculation of the helicopter's relative position at any given time.

2.1.1 T-Rex 600 E Super Pro Remote Controlled Helicopter

The RC helicopters used for this project were the T-Rex 600 E Super Pro model, produced by Align Corporation. The helicopters were purchased as a kit and were assembled at the start of the project. Due to its relatively large size (specifications in Table 1), the helicopter

provided stability uncommon in smaller helicopter models. Shown in Figure 10, the T-Rex 600 E Super Pros are battery powered, giving a flight time of eight to ten minutes per charge.

Table 1. T-Rex 600 E Super Pro specifications.

Parameter	Measurement
Length	1160 mm (45.6 in)
Height	353 mm (13.9 in)
Width	210 mm (8.27 in)
Main Blade Length	600 mm (23.6 in)
Main Rotor Diameter	1347 mm (53.0 in)
Tail Rotor Diameter	260 mm (10.2 in)
Empty Weight	2350 g (5.2 lbs)
Loaded Underway Flight Weight	5805 g (12.8 lbs)

Figure 10. T-Rex 600 E Super Pro remote controlled helicopter.

Prior to the first underway flight with the helicopters, a flotation system was developed and tested to ensure that the rotorcraft could be recovered in the case of a water landing. The initial design, shown in Figure 11, utilized a pair of Styrofoam pontoon-style floats strapped to each of the helicopter's landing skids. Because the Styrofoam had the tendency to grab the YP's nonskid deck surface, this design made the helicopter harder to control when flying close to the flight deck during takeoffs and landings. The second flotation system design, which was ultimately used during all underway flight operations, incorporated Styrofoam spheres attached to a pair of "spider legs" mounted underneath the helicopter. The "spider legs" with free spinning

whiffle balls attached at the ends of each leg are commonly used by beginner model helicopter pilots in order to mitigate the risk associated with handling the helicopter low to the ground. This flotation system is depicted in Figure 12.

Figure 11. T-Rex 600 E Super Pro with initial pontoon-style flotation system

Figure 12. T-Rex 600 E Super Pro with "spider leg" flotation system.

2.1.1.1 Remote Controlled Helicopter Training and Safety Program

Safe operation of the RC helicopter was of utmost importance during this project. A number of steps and precautions were taken in order to minimize the risk associated with the underway testing, and, similar to how the U.S. Navy performs risky operations, a controlled

buildup to underway testing occurred. The primary helicopter pilot, MIDN Jason Metzger, started flying computer simulations of the model helicopter in the spring of 2011. Once the helicopter models were constructed for this project, he progressed to practicing basic hovering and flying simple paths with the models that would eventually be used underway. Once deemed proficient in basic helicopter handling over the land, the next step was hovering over the flight deck of YP676 when moored to the pier. At the end of November 2011, the first underway flights were flown, starting with a hover underway and then progressing to basic pattern flying with the ship maintaining a specific wind over deck condition. In January of 2012, a civilian RC helicopter pilot, Mr. Charles Rosario, joined the project as a mentor to help tune-up the helicopters and assist with some of the underway flying.

Onboard YP676, a number of physical precautions were taken to ensure the safety of the research team, ship's crew, and the ship itself during flight operations. Whenever the helicopter was powered up, nobody was allowed on the flight deck of the ship, and the pilot and safety observer were shielded behind a sturdy metallic crash shelter. The controller for the helicopter was equipped with a "kill switch" in case the aircraft got out of control, and the flotation system was always attached in case a loss of power occurred when the helicopter was over the water.

No safety infractions occurred over the course of the underway testing. Control of the helicopter was lost once due to a mechanical failure when the helicopter was approximately 120 feet aft of the ship, but no personnel were in danger during this mishap or the recovery of the helicopter.

2.1.2 x-IMU Inertial Measurement Unit

The x-IMU, produced by x-io Technologies, was the instrument chosen to collect data pertaining to the helicopter's in-flight orientation and disturbances. The x-IMU, shown in Figure 13, features an assortment of on-board sensors powered by an attached lithium-polymer battery and saves all data to an enclosed Micro SD card, allowing the IMU to operate independently of a computer or external power source during flight operations. For this project, data recorded by the x-IMU's triple axis 16-bit gyroscope and triple axis 12-bit accelerometer were analyzed.

Figure 13. x-IMU Inertial Measurement Unit

2.1.2.1 x-IMU Waterproofing

Special care was taken to protect the x-IMU from water during the underway flight sessions. During flight operations in the Chesapeake Bay, the helicopter is exposed to sea spray and the risk of a water landing. Because of the importance of the x-IMU's reliability to this project, the IMU was secured inside of an Otterbox® waterproof case, which was then attached to the base of the helicopter. This waterproofing system was tested prior to going underway.

2.1.2.2 x-IMU Calibration

In order to understand what the disturbed helicopter flight paths would look like in the IMU's data, a number of sample flights over land were conducted prior to going underway. During these flights, the pilot would add extraneous control inputs to simulate the helicopter's expected motion in the air wake. This process provided sample data to help distinguish between steady helicopter flight and disturbed helicopter flight when analyzing the data from underway operations.

2.1.3 SD GPS Data Logger

In order to effectively map the flight of the helicopter, two separate SD GPS Data Loggers, as shown in Figure 14, were used. The GPS units, produced by OHARARP, LLC, are powered by an on-board lithium polymer battery, record GPS data at a 10 Hz sampling rate, and write all information to an embedded Micro SD card. Two separate units were used during the project: one was mounted on the helicopter, and one stayed onboard the YP as it moved through the water in order to provide a reference position.

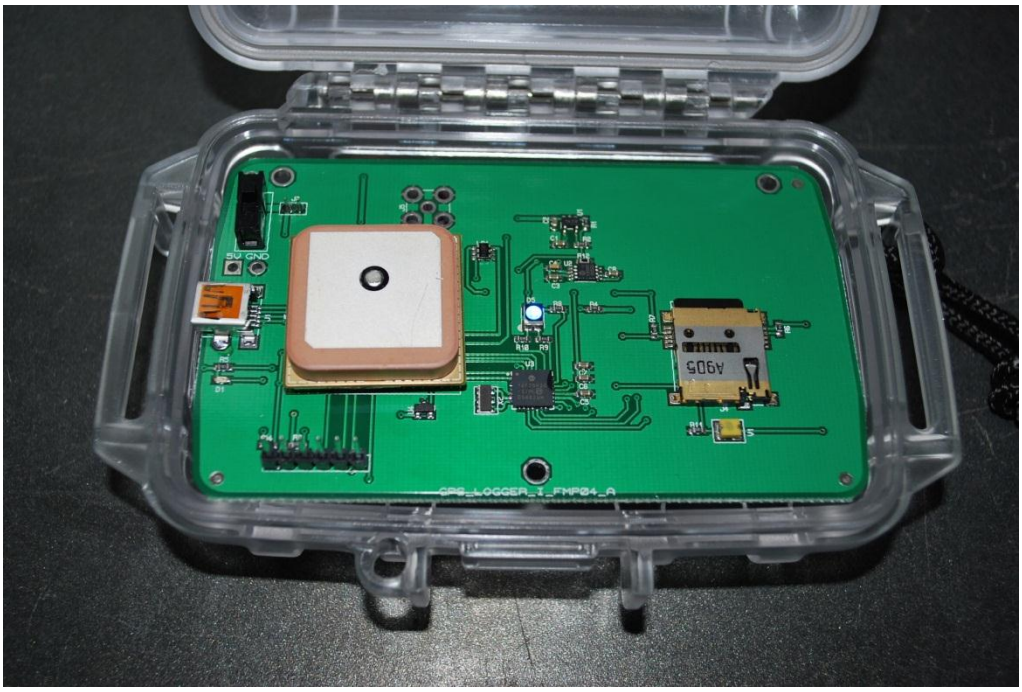

Figure 14. SD GPS Data Logger.

2.1.3.1 SD GPS Data Logger Waterproofing

Similar to the x-IMU, the SD GPS Data Logger was waterproofed inside of an Otterbox® case in order to protect it against sea spray and the possibility of submersion during underway operations. The Otterbox® was then externally attached to the helicopter to provide easy access to the GPS unit and so that the helicopter's frame would not interrupt the line-of-sight between the GPS receiver and the satellites in orbit.

2.1.3.2 SD GPS Data Logger Calibration

Prior to the first underway data collection session, the SD GPS Data Logger was tested to ensure its accuracy and to identify the need for any calibration. GPS receivers commonly give latitude and longitude positions, but the accuracy of this data can vary based on atmospheric conditions and the amount of satellites visible at a given time. For this project, the heading and speed data from the GPS were used to compute the helicopter's flight position at any given time. Even if the GPS sensor does not accurately know its position, it can give a very accurate speed and heading output based on measuring the Doppler shifts from the satellites in orbit, whose paths are very precisely known.

To test the accuracy of the GPS, each unit was attached to one of the helicopters during a number of practice flights. After the flying was done for the day, the speed and heading data was integrated and compared with the measured results of known takeoff and landing positions. It was found that, over the course of each flight, the GPS derived position data was accurate to within one meter.

2.2 System Integration

With the instrumentation discussed, the velocity components of the air wake cannot be directly measured in the manner of testing already completed over the flight deck. Instead, the interaction between the helicopter and regions of different air wake velocity were recorded. Figure 15 shows a CFD analysis of a YP's air wake with the wind crossing the flight deck at 60° off of the starboard bow. The image illustrates that the airflow immediately around the flight deck is only one area of concern for helicopters operating in the vicinity of a naval vessel. The white dashed lines indicate landing approaches commonly flown by different services: the approach that arrives straight in from the stern of the ship is used by the United States Navy, and the approach having the helicopter pull up even with the ship and moving laterally onto the flight deck is used by the Royal Navy. Regardless of the approach path taken, a helicopter will experience regions of velocity change on the way to making a successful landing. Using the proposed instrumentation, the effects that these large regions of velocity change have on the rotorcraft can be measured and mapped.

As the helicopter flies in and out of these large regions of the air wake where the velocity is predicted to change with time, it experiences noticeable disturbances in its flight. These disturbances have been noticed as periodic roll and pitch rates and dramatic acceleration changes in the vertical axis. Qualitatively, these reactions have been observed visually by personnel watching the flight operations and can be seen on video recordings of each flight. The pilot for each underway session also noted times when he "felt" disturbances with the helicopter when trying to fly it steadily through the air wake. Quantitatively, these disturbances have been recorded by the accelerometers and gyroscopes on the x-IMU. When paired with the derived GPS position data, this method allows for the mapping of the helicopter's flight disturbances, resulting in an effective mapping of the air wake's effects on the helicopter.

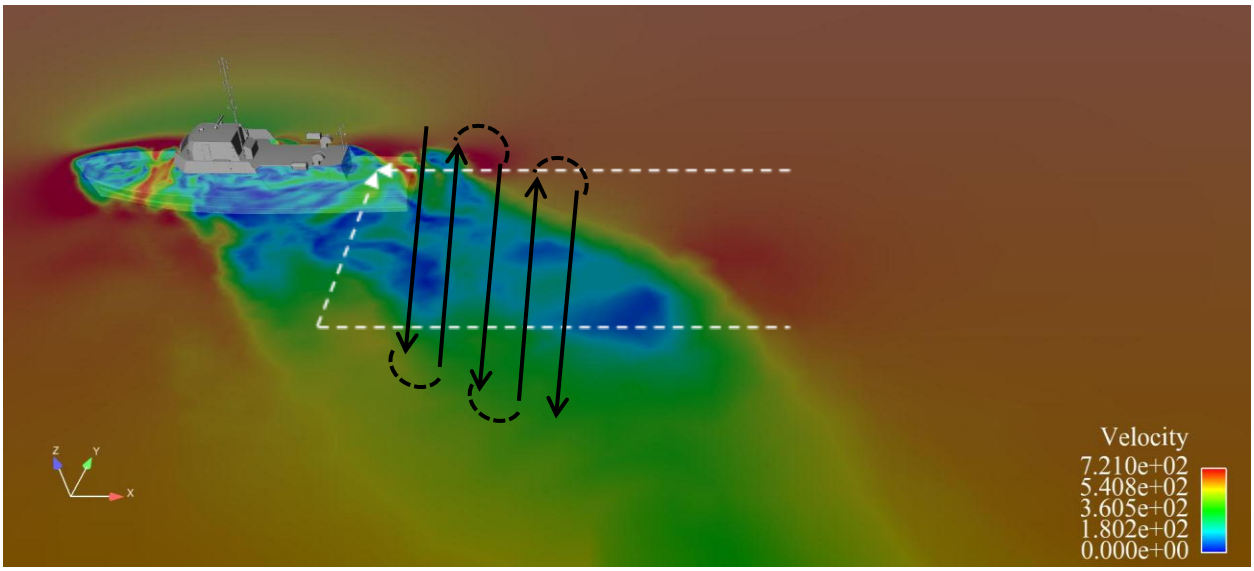

Figure 15. YP Air Wake for a 60° crosswind off the starboard bow. Black arrows represent a sample RC Helicopter flight profile to detect variation in ship air wake. White arrows represent common fleet helicopter approach paths.

2.3 Data Collection

A total of nine underway flight sessions were conducted in coordination with this project. Accomplishments varied between each underway, beginning with systems checks and flight practice on YP676 and concluding with multiple data-collection flights per underway session. These underways are detailed in Table 2.

Before each underway, a pier-side systems check was conducted to ensure proper helicopter operation prior to embarking onto the Chesapeake Bay. Typically, the pilot tested the helicopter in a hover over the ship's flight deck and then demonstrated one or two sample departures and approaches from the flight deck. Following this short test, YP676 would get underway. As the ship transited down the Severn River, the x-IMU to be mounted on the helicopter and the two SD GPS Data Loggers were initialized, tested, and synced in time. The charge levels on each of the helicopter's lithium polymer batteries were also checked, and any deficient battery was plugged into one of the two battery chargers brought onboard the ship for the project. This process was completed at approximately the same time the YP had reached the Chesapeake Bay and had maneuvered to be ready for the first data collection flight.

Table 2. Detailed schedule of underway flight operations.

Date of Underway	Number of Flights Conducted	Notes
November 29, 2011	0	First Underway Systems Check
December 2, 2011	4	Underway Hover and Approach Practice
January 6, 2012	5	3 Flights at β=0° 1 Flight at β=15° 1 Flight at β=45°
January 13, 2012	1	Very Rough Seas
January 20, 2012	2	2 Flights at β=15°
February 3, 2012	6	4 Flights at β=30° 2 Flights at β=15°
February 10, 2012	5	5 Flights at β=15°
February 17, 2012	1	Mechanical Failure During First Flight Stopped Operations for the Day
March 2, 2012	4	2 Flights at β=15° 2 Flights at β=30°

For each data collection flight, the YP's craft master steered the ship in order to produce a desired wind-over-deck setting based on measurements from the bow reference anemometer. It is common practice for naval surface ships to maneuver in a manner to produce a certain wind-over-deck condition when conducting flight operations. These conditions can be requested by the pilot or can be specified by launch and recovery envelopes. Once the ship had settled on a course and speed to produce the correct wind settings, flight operations proceeded. During each flight, slight variations in the wind direction occurred based on the ambient currents and winds in the testing area; this cannot be avoided due to the nature of the testing.

At the start of each flight, the pilot installed the helicopter's batteries, did one final systems check, and moved to his position to conduct flight operations. Depending on the altitude level of the data being collected, the pilot stood either on the fantail of the YP above the flight deck or along one of the passageways on the port side of the ship. The video camera operator would then start filming and place a watch in front of the camera's lens. This watch was synced with the x-IMU and the SD GPS Data Logger units, allowing the video to be used as a means of interpreting the data obtained during each underway flight. Each flight typically lasted between six and nine minutes, depending on the weather and the battery charge. During the flights, the pilot attempted to fly the helicopter slowly and steadily in paths parallel to the aft end of the ship

through the expected air wake region. This methodology is depicted by the black arrows in Figure 15. Data collection flights were focused on wind settings of 15° off of the starboard bow and 30° off of the starboard bow (β=15° and β=30°, respectively).

After the final scheduled flight for each underway session, the Micro SD cards were taken out of x-IMU and the SD GPS Data Loggers to have the data verified before returning to the Naval Academy. The data was not processed and analyzed at this time; instead, this step ensured that all of the equipment functioned properly. All of the equipment was then taken back to the Naval Academy where batteries were recharged and mechanical and programming issues were addressed before the next underway.

2.4 Data Reduction

Three different types of data were obtained for each flight, and the combination of each piece of information is required to analyze the information gained from the underways. Gyroscopic rates and acceleration data from the x-IMU onboard the helicopter allowed for the detection of the helicopter's interaction with the ship's turbulent air wake. Heading and speed data from the SD GPS Data Logger units onboard both the YP and the helicopter were integrated to calculate the position of the helicopter with respect to the flight deck at any moment. The video recordings of each flight allowed for review of the data in the context of each underway flight session, enabling viewers to distinguish between measured gyroscopic rates caused by the air wake or rates inadvertently induced by the pilot during flight.

2.4.1 Flight Videos

The first step in the data reduction process was trimming the flight videos down to the duration of each flight and adding a timestamp to each clip. This was done using Adobe Premiere Elements 9, a commercial video editing software suite. A screenshot from an edited flight video is shown in Figure 16. Since the watch shown at the start and end of each flight was in sync with the clock onboard the x-IMU and the GPS units, the video timestamps could be used to determine at what times to start and stop analyzing air wake disturbance data. By reviewing the flight videos, a table similar to Table 3 was populated for each flight, showing the times that the helicopter was flying through the YP's air wake with minimal control inputs being added by the pilot.

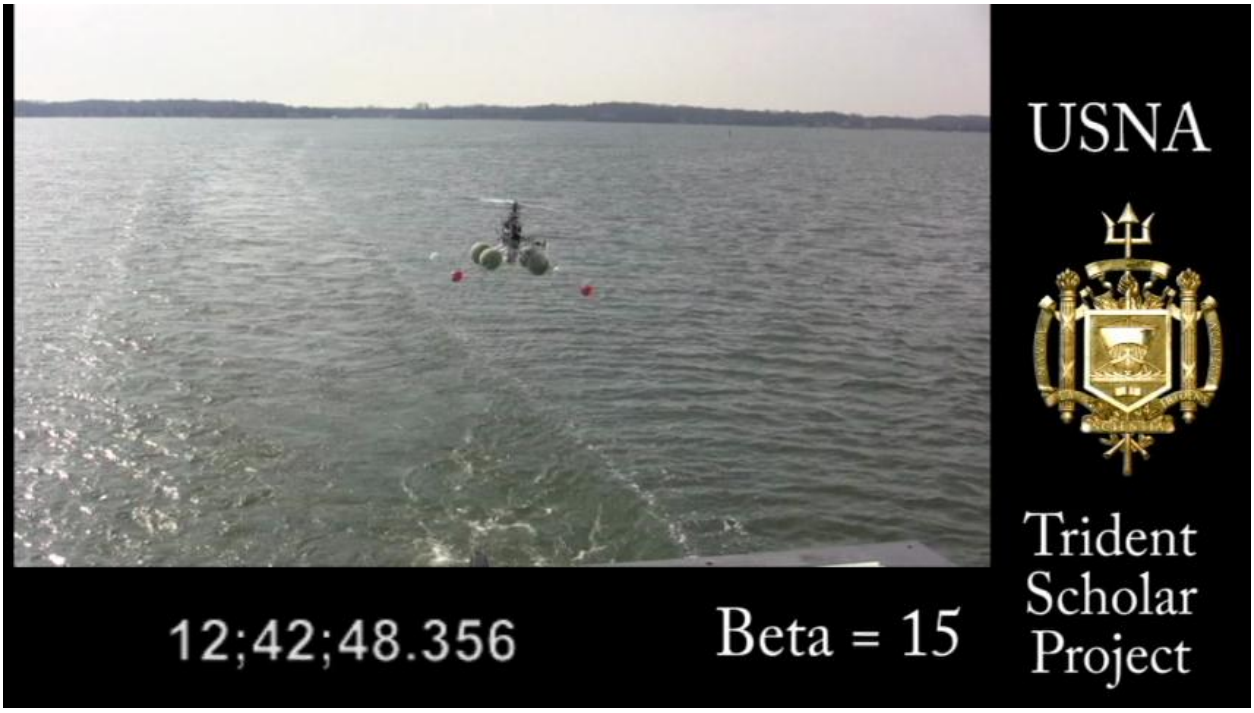

Figure 16. Screenshot of edited flight video from February 10, 2012.

Table 3. Minimum pilot input times for a flight on February 10, 2012.

Minimum Pilot Input Passes					
Start Time			**End Time**		
hh	**mm**	**ss**	**hh**	**mm**	**ss**
Flight1					
12	44	8	12	44	16
12	45	4	12	45	15
12	45	30	12	45	43
12	45	51	12	46	13
12	46	31	12	46	42
12	46	53	12	47	2
12	47	15	12	47	25
12	47	45	12	47	54
12	48	14	12	48	25
12	48	36	12	48	43

2.4.2 GPS Data

The next step was to use the GPS data to derive the position of the helicopter with respect to the ship at all times during each flight. As previously discussed, the actual latitude and

longitude positions recorded by the GPS vary in accuracy by a few meters. Instead, the heading and speed outputs from the SD GPS Data Logger, which are accurately calculated using an onboard algorithm that compares Doppler shifts from the orbiting GPS satellites, were integrated to determine the ground tracks of the helicopter and the YP during each flight. Equations (1) through (4) below describe the integration process for determining ground path position of the helicopter from the heading and speed data. The same equations are used for the helicopter. All of the calculations regarding the GPS data were completed using the Relative Position Calculator from GPS Data (GPScrunch2.m) MATLAB script found in Appendix B.

$$V_{i_{x,heli}} = \bar{V}_{i_{heli}} \sin\left(\beta_{i_{heli}}\right) \tag{1}$$

$$V_{i_{y.heli}} = \bar{V}_{i_{heli}} \cos\left(\beta_{i_{heli}}\right) \tag{2}$$

$$D_{i_{x,heli}} = D_{i-1_{x,heli}} + V_{i_{x,heli}} \Delta t \tag{3}$$

$$D_{i_{y,heli}} = D_{i-1_{y,heli}} + V_{i_{y,heli}} \Delta t \tag{4}$$

The ground tracks were then matched up using the common time sync to calculate the relative position of the helicopter with respect to the YP at any given time. By using the law of cosines, the relative distance and azimuth angle between the YP and the helicopter were calculated as detailed in equations (5) through (8). Equations (9) and (10) convert this into simple two-variable coordinates for mapping these locations aft of the flight deck.

$$a_i = \sqrt{\left(D_{i_{x,heli}} - D_{i_{x,YP}}\right)^2 + \left(D_{i_{y,heli}} - D_{i_{y,YP}}\right)^2} \tag{5}$$

$$b_i = \sqrt{\left(D_{i_{x,YP}} - D_{i-1_{x,YP}}\right)^2 + \left(D_{i_{y,YP}} - D_{i-1_{y,YP}}\right)^2} \tag{6}$$

$$c_i = \sqrt{\left(D_{i-1_{x,YP}} - D_{i_{x,heli}}\right)^2 + \left(D_{i-1_{y,YP}} - D_{i_{y,heli}}\right)^2} \tag{7}$$

$$\theta_i = \cos^{-1}\left(\frac{{a_i}^2 + {b_i}^2 - {c_i}^2}{2a_i b_i}\right) \tag{8}$$

$$D_{i_{x,relative}} = a_i \sin(\theta_i) \tag{9}$$

$$D_{i_{y,relative}} = a_i \cos(\theta_i) \tag{10}$$

In these equations, a_i is the relative distance between the YP and the helicopter at time i, and θ_i is the relative angle between the YP and the helicopter (0° is straight aft of the ship).

Through this integration scheme, the position of the helicopter relative to the flight deck at every time interval becomes known. The SD GPS Data Logger records data at a rate of 10 Hz, which is much faster than major deviations in speed and course of either the helicopter or the YP, giving good confidence to the derived position data. An example of a helicopter's flight path integrated using this method is displayed in Figure 17.

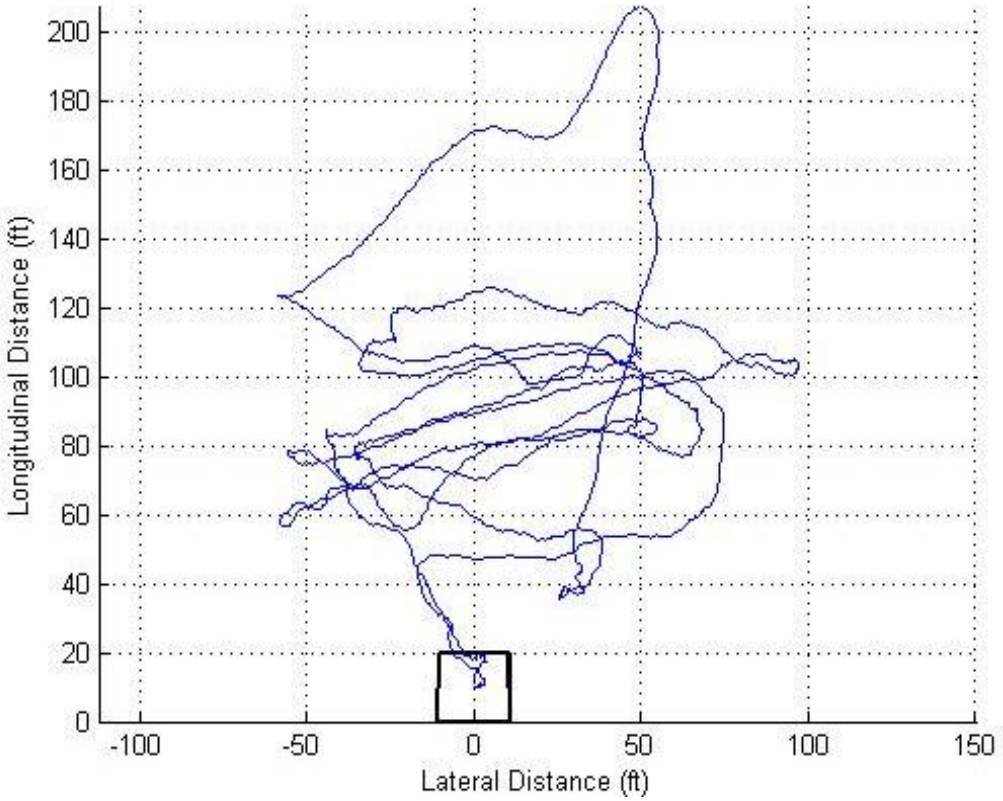

Figure 17. GPS derived flight path aft of the flight deck of YP676.

This flight path shows the helicopter flying in the manner suggested in Figure 15. Review of this and other sets of data showed that many of the earlier underway flights occurred 60 to 120 feet aft of the flight deck. Later flights were tailored to explore the regions closer to the ship.

2.4.3 IMU Data

The data outputs from the x-IMU were saved onto the Micro SD card onboard the unit. After being extracted using software provided by x-io Technologies, a set of nine .csv files were available for analysis. A number of these files, such as the ones that contained battery and

thermometer data, were ignored for this project. Only three of the sets of data were used for analysis: date and time registers, gyroscopic rates, and accelerations.

During the underway flights, the x-IMU recorded acceleration and gyroscopic data at a rate of 128 Hz. Because of the large size of the data files (approximately 110 MB per underway session), the data had to be separated manually by flights into separate files before being imported into MATLAB for analysis. The first MATLAB script for the IMU data, IMU Data Trim and Repackage Code (IMUunpack_V2.m), took the .csv data for each flight, trimmed it down to the start and end times of each flight recorded from watching the underway flight videos, and saved it as a .mat (MATLAB data) file for each flight. Each of these files takes up approximately 2.5 MB of space and greatly reduces the processing time for further calculations by freeing up system memory allocated to MATLAB.

A separate MATLAB script was used to create streaming video files of the gyroscopic and accelerometer data for each flight. Raw Data Video Creator (HeliMovieMaker_V2.m) imports the data files saved by IMU Data Trim and Repackage code and processes ten seconds of data at a time. The code then advances the data by ten time steps and creates another plot, eventually saving all of the produced plots to a single .avi video file to be compared directly against the videos of the underway flights. The title at each time increment included the time stamp so that the plotted data could be synced with the flight videos. Figure 18 shows an example screenshot from one of these raw data videos.

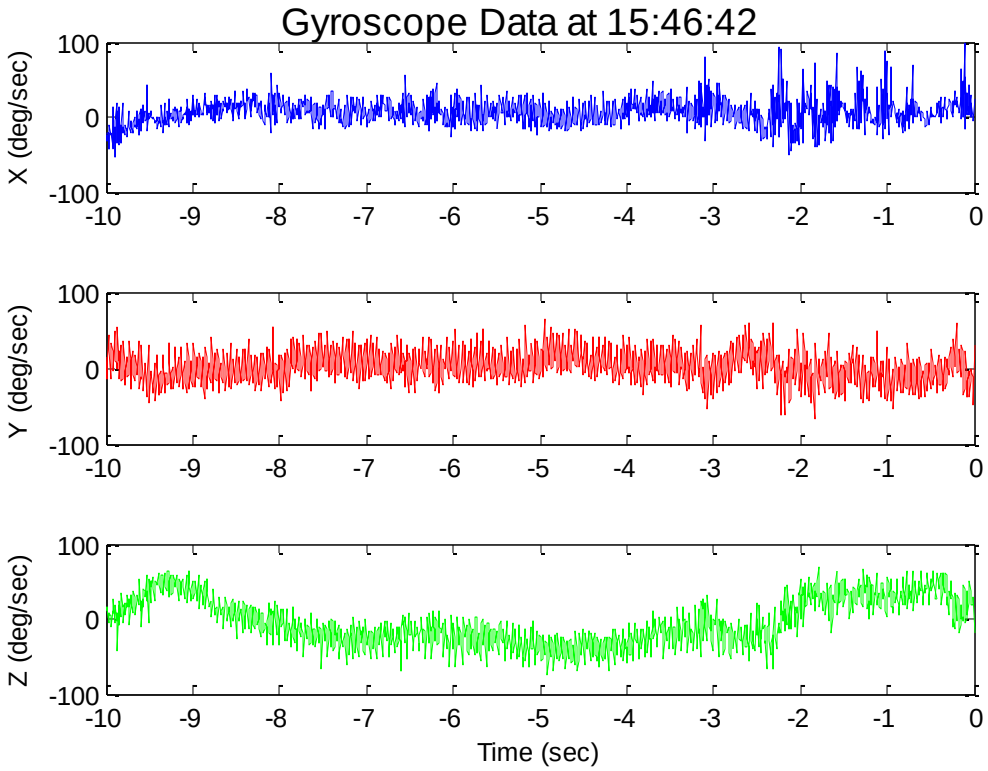

Figure 18. Raw gyroscopic data plot.

A box filter was applied to each of the gyro axis to eliminate the electrical noise inherent in both the IMU and the helicopter. The filtered quantity can be obtained by the convolution integral in equation (11)[14].

$$\tilde{f}(x,t) = \int_{0}^{\infty} G(r,x) f(x-r,t) dr \tag{11}$$

In this equation, *G* is the filter kernel (a box filter for this study) and *f* is the raw data to be filtered. Figure 19 gives an example output of the filtered data (red line) superimposed over the raw data (gray background).

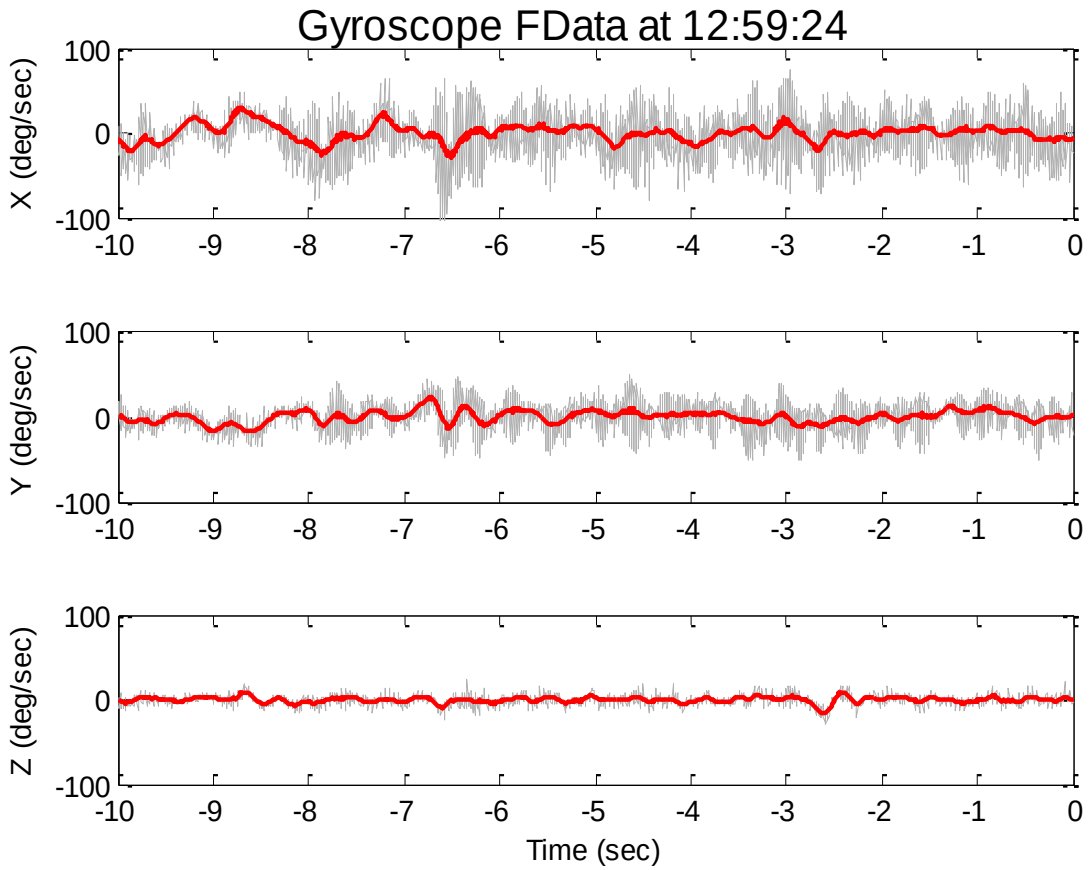

Figure 19. Filtered gyroscopic data plot.

2.4.4 Data Correlation

Because all three sources of data were synced to the same clock, they can be correlated and used together to determine at which locations aft of the ship the helicopter felt disturbances caused by the YP's turbulent air wake. The data correlation process employed for this project is simple but currently requires a man-in-the-loop to match and interpret the data.

Starting with the times of minimal pilot inputs for each flight, such as what is displayed in Table 3, the filtered data videos are viewed to determine at what moments within the times of minimal inputs the helicopter shows a reaction that compares to what is expected in the air wake. The standard for this comparison is qualitative, based on observations over the nine underway sessions and the x-IMU calibration and practice flying off of the ship.

Analyzing the filtered data videos showed that the gyroscopic rate data more clearly showed when the helicopter was being affected by the air wake than the accelerometer data. Additionally, since the T-Rex 600 E Super Pro is built with a heading-lock gyro which controls

the pitch of the tail rotor blades to properly counter the torque produced by the main rotor blades, the gyroscopic rates in the Z-axis (yaw rate) were ignored. Thus, only the gyroscopic rates in the X-axis (roll rate) and the Y-axis (pitch rate) were used to determine the times at which the helicopter was under the influence of the ship's air wake.

Figure 20 depicts an instant where the gyroscopic data shows the helicopter entering a region where its flight path is disturbed from normal, steady flight. Before the helicopter reaches the sharp velocity gradient at the edge of the YP's air wake, the pitch and roll rates measured by the x-IMU's gyroscopes remain relatively constant. As shown in Figure 20, once the helicopter reaches the sharp velocity gradient, there is an impulsive pitch and roll rate measured by the gyroscopes. This instant is indicated by the dashed line in the figure. While the helicopter remains inside the air wake, varying pitch and roll rates are measured until the helicopter either departs the air wake or the pilot adds control inputs to maneuver the helicopter.

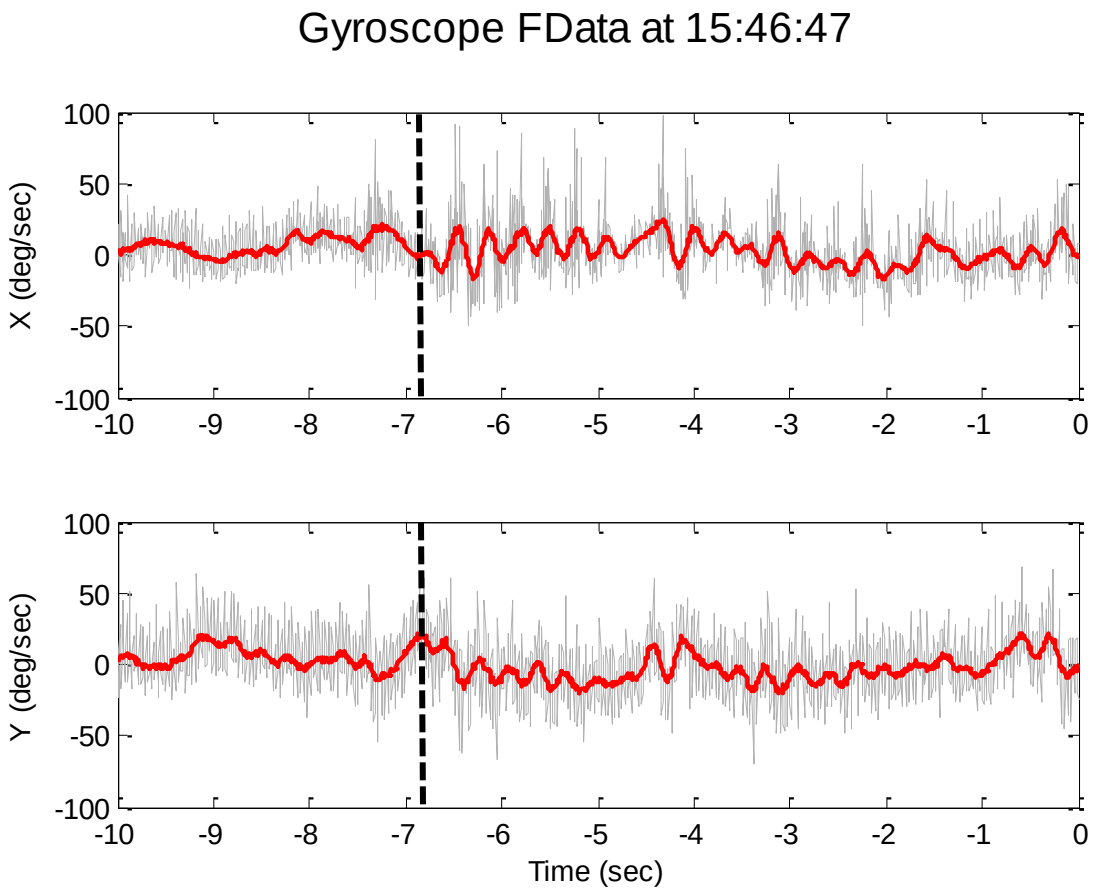

Figure 20. Pitch and roll gyroscopic data along a flight path into the air wake. Dashed line indicates time at which the helicopter entered the wake.

Visual observations showed that the helicopter appeared to experience a Dutch Roll in the ship's air wake. A Dutch Roll is a characterization of an aircraft's motion when in the presence of a slip. When a component of the freestream velocity is not in the same direction as the flight path, the helicopter oscillates about its path sinusoidally in pitch, roll, and yaw.[15] For the T-Rex 600 E Super Pro helicopter with the heading-lock gyro, this sinusoidal yaw rate is not observed, but the pitch and roll rates are measured by the x-IMU. Often, this motion is damped, and pilot intervention will usually eliminate this motion over prolonged periods of time. However, because of the manner in which the flight operations were conducted and the understood nature of the ship's turbulent air wake, the helicopter appeared to be in a slip whenever it was in the air wake region because the slip angle and velocity were constantly changing.

By reviewing each flight data video alongside the times for each flight similar to that contained in Table 3, another set of times was developed for each flight. Table 4 corresponds to the same flight as Table 3 but shows the start and end times for which the x-IMU data showed disturbances in the helicopter's flight potentially caused by the YP's air wake. Some of the rows in the table indicate that the gyroscopic data did not conclusively show a clear beginning or end to the helicopter's oscillatory motion in the air wake. These time omissions were commonly found on each flight.

Table 4. Times in air wake as indicated by gyroscopic data.

Air Wake Effects on IMU					
Start Time			**End Time**		
hh	**mm**	**ss**	**hh**	**mm**	**ss**
Flight1					
12	44	13	12	44	16
air wake interaction not found on IMU					
12	45	36	12	45	41
12	45	58	12	45	4
12	46	37	12	46	40
12	46	56	12	47	1
12	47	20	12	47	21
12	47	48	12	47	53
air wake interaction not found on IMU					
12	48	37	12	48	40

The start and end times taken from the IMU data detailing the helicopter's reactions were then used with the calculated relative distances between the helicopter and the YP from equations (9) and (10) to map the locations relative to the YP where the helicopter showed disturbances in its flight due to the ship's air wake. These mappings were overlaid with results from CFD simulations already completed for the Ship Air Wake project, such as the images displayed in Figure 21 through Figure 24. Each flight's results are sorted by the relative wind over deck angle, β. This angle, along with the approximate altitude of the helicopter during each flight, connects each flight data set and CFD simulation. These graphics allow for the direct comparison of the location of the measured helicopter-air wake interactions and the location of the air wake as proposed by CFD simulations.

For each of the following visualizations produced from the CFD simulations, the data presented is at a singular horizontal level. The CFD model of the YP, which contains approximately 15 million tetrahedrons, calculates air velocity data at every altitude specified within the model, from sea level to at least four times the height of the YP. In order to map the air wake, horizontal slices of the CFD data had to be taken to compare to each approximate altitude flown during underway testing. Multiple altitudes cannot be directly compared.

In all of the CFD visualizations, the colors indicate the magnitude of the air velocity at each location. Orange indicates the ambient air condition. Regions where colors shift towards red are where the air flow is expected to increase in speed, and colors shifted towards yellow, green, and blue indicate a slower air velocity.

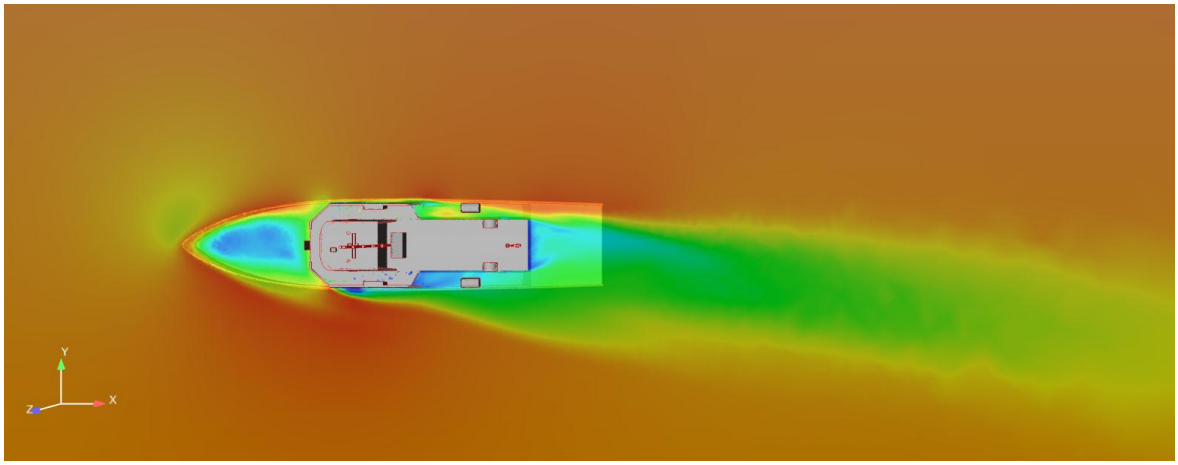

Figure 21. CFD simulation for β=15° at a height of the top of the hangar.

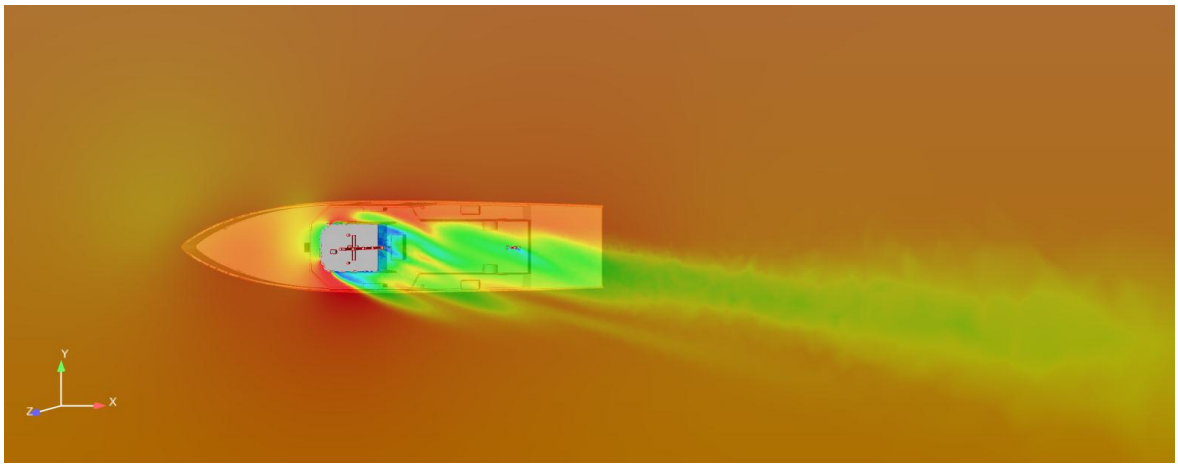

Figure 22. CFD simulation for β=15° at a height of the top of the conning tower.

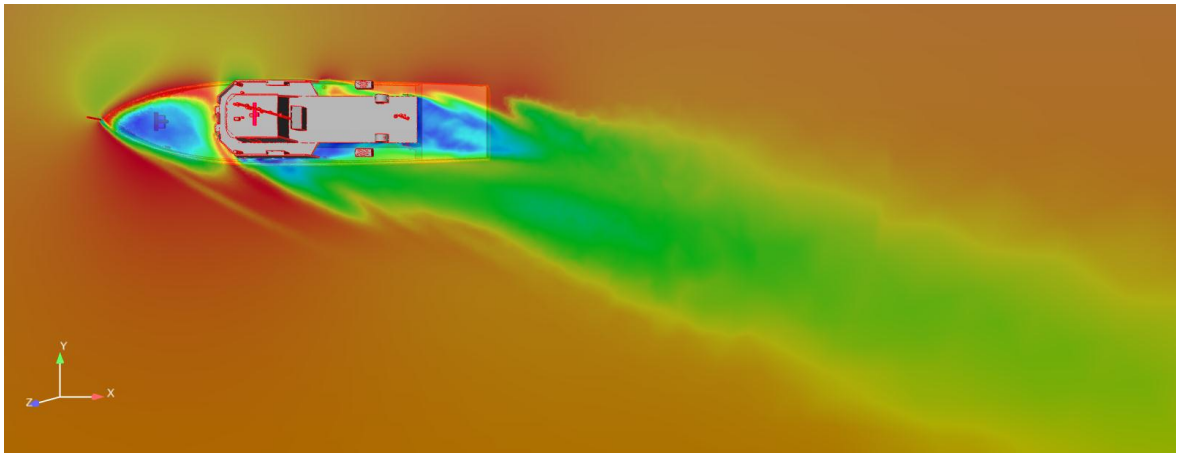

Figure 23. CFD simulation for β=30° at a height of the top of the hangar.

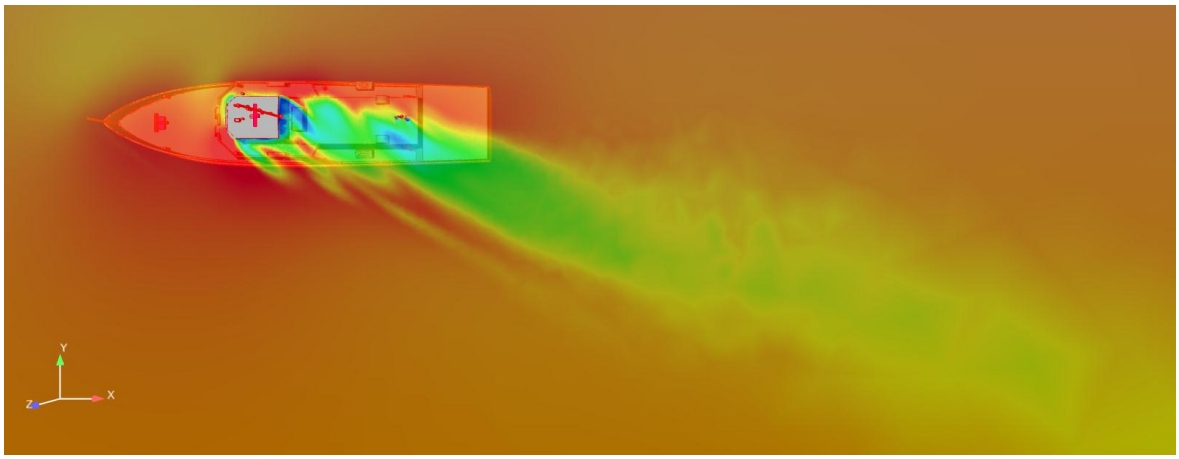

Figure 24. CFD simulation for β=30° at a height of the top of the conning tower.

3 Results

The majority of the underway flights were completed at β=15° and β=30° conditions. Flow visualizations from previously conducted CFD analyses of both of these settings were produced in order to compare the underway flight data with CFD predictions.

3.1 Top of Hangar Structure with β=15°

One of the two altitudes available for comparison studies with the CFD simulations is at the height of the top of the hangar structure immediately before the flight deck on YP676 (approximately 2 meters (6 feet) above the flight deck). During flight operations, the pilot attempted to keep the helicopter consistently at this height for each lateral pass through the ship's air wake while minimizing control inputs; however, deviations of ±1 meter (approximately 3 feet) from the desired elevation were commonly observed, particularly upon entering and exiting the air wake.

Based on the method previously discussed, the GPS locations were mapped when it was determined through visual inspection and IMU data that the helicopter was experiencing disturbances in its flight due to the turbulent air wake created by the ship. These results can be seen in Figure 25. In this image, the data obtained from the underway flight operations indicating flight path disturbance due to ship air wake is presented in dashed blue lines, whereas the CFD simulations compose the background image. Orange is the freestream velocity for the CFD simulation, and the colors shift towards green and blue for slower air velocities.

Figure 25 shows a good correlation between the location of the YP's air wake from the CFD simulation and what was measured by the instrumentation onboard the helicopter during testing. As the data gets further aft of the ship, the air wake measured during the helicopter flight operations appears to suggest that the ship's air wake should drift to the port side of the ship more so than what the CFD simulation predicts. Close to the aft end of the ship, though, the data gathered during flight operations corresponds well with the CFD simulation.

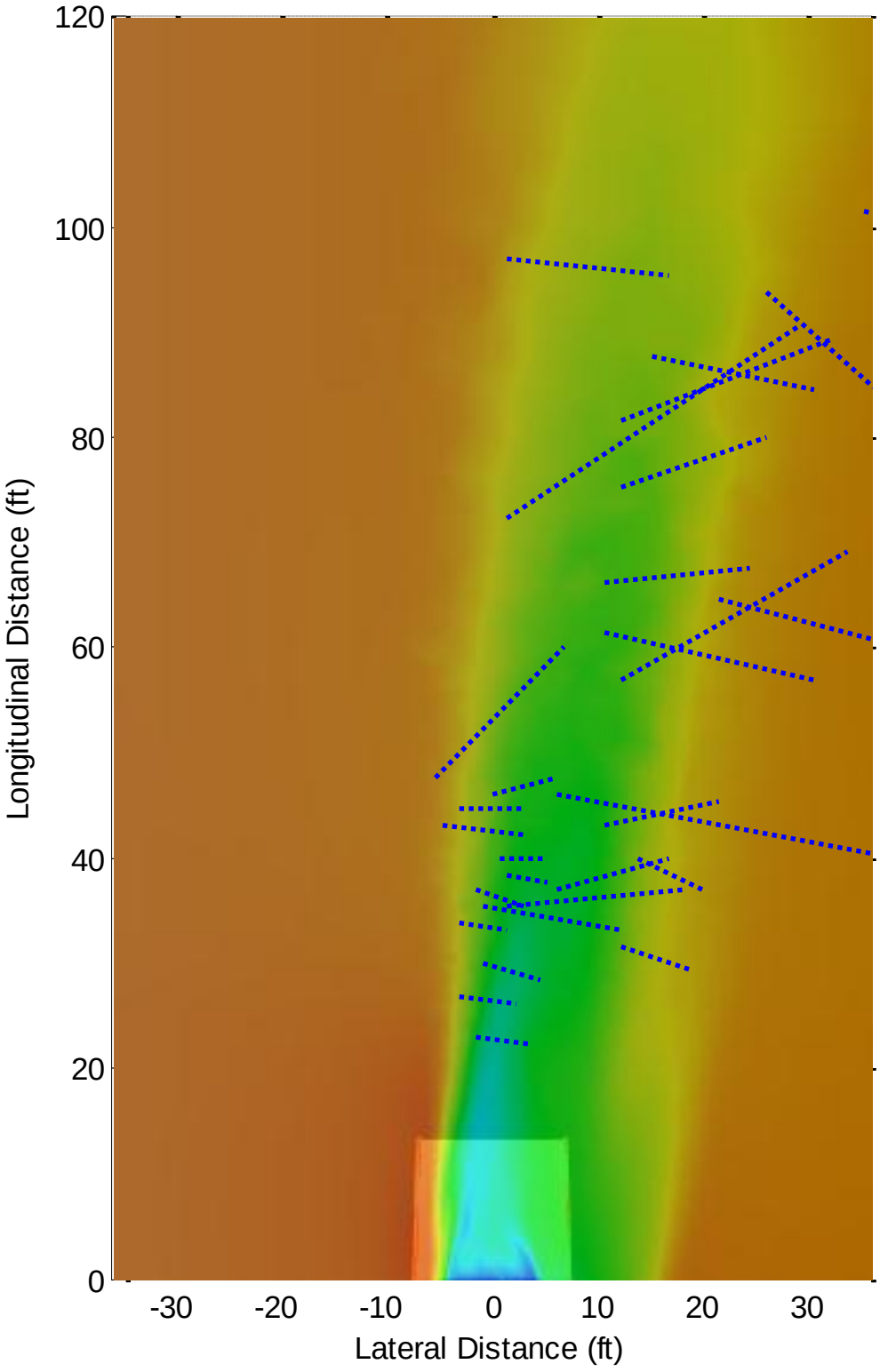

Figure 25. Measured air wake location (blue dashed lines) and CFD simulation (colored background) for β=15° at the top of the hangar structure.

A change in flight operation styles can be visualized in Figure 25. During the first two underways where data units were attached to the T-Rex 600 E Super Pro, the helicopter would start a lateral pass in a hover to the starboard side of the wake and slowly move parallel to the aft end of the ship towards the air wake. Once the helicopter appeared to enter the air wake, the helicopter suffered a noticeable immediate loss in altitude, and the pilot would add control inputs and maneuver the helicopter back to the starboard side of the ship. It appeared as if the starboard edge of the air wake could be characterized by a downwash (later testing showed that the port side of the ship had characteristics of an upwash). This method of flying was designed to effectively map the starboard boundary of the YP's air wake. This boundary was very apparent visually to the pilot and observers during testing and appears in the figure as a series of shorter dashed-line segments on the starboard edge of the wake near the aft end of the ship.

For the final two underway sessions, after reviewing the gyroscopic IMU data from previous flights, it was decided that the pilot should attempt to fly the helicopter slowly through the entirety of the wake with minimal control inputs. By flying the helicopter in this manner, the gyroscopic data would clearly show at what times the helicopter entered and exited the ship's air wake. While it was in the air wake, the IMU measured more rapid periodic pitching and rolling on the helicopter, and it is relatively clear in the data when the helicopter enters and exits the turbulent wake. These measurements are characterized by the longer segments of dashed lines in Figure 25. However, this manner of flying is much more challenging for the pilot. When the helicopter reacts to velocity gradients in the wake, the natural response is to add control inputs to the system to keep the helicopter stable. It proved to be very difficult to fly the helicopter consistently without adding control inputs.

During the underway flight operations, the YP's craft master attempted to keep the ship under the same wind over deck condition. This information is provided to the craft master via the bow reference anemometer shown in Figure 6. During testing, though, it can be very difficult to keep the ship at a constant wind over deck condition. As the winds and currents shift, the ship has to react to keep the relative wind constant. This could explain the apparent drift of the measured wake towards the port side further aft of the ship. On a couple of occasions, it was noted that the relative wind angle had indeed increased from $\beta=15°$ to between $\beta=30°$ and $\beta=45°$ on the same flight. Once this had been noticed, the craft master would alter course and speed to regain the desired wind over deck condition. The drift towards increasing relative wind angles

would cause the wake to shift more towards the port side of the ship, as suggested by some of the data.

3.2 Top of Conning Tower with β=15°

The other altitude flown during underway testing was at the level of the top of the conning tower, approximately 3 meters (10 feet) above the level at the top of the hangar structure. Due to cautions taken by the pilots, a substantial amount of the initial flights took place at this altitude. Corrective actions were taken to keep the pilot flying at the lower height for later underway sessions.

Figure 26 displays the helicopter's measured reactions to the ship's air wake for the β=15° case when the helicopter is at the height of the conning tower. Many of the trends that were seen on Figure 25 for the hangar height situations are seen in this figure as well. Further aft of the ship, the measured air wake seems biased to the port compared with CFD analysis, possibly due to the variable wind over deck conditions present during underway testing. Overall, the measured air wake reactions correlate well with the CFD analysis of the air wake locations, but not as well as the measurements at the top of the hangar level.

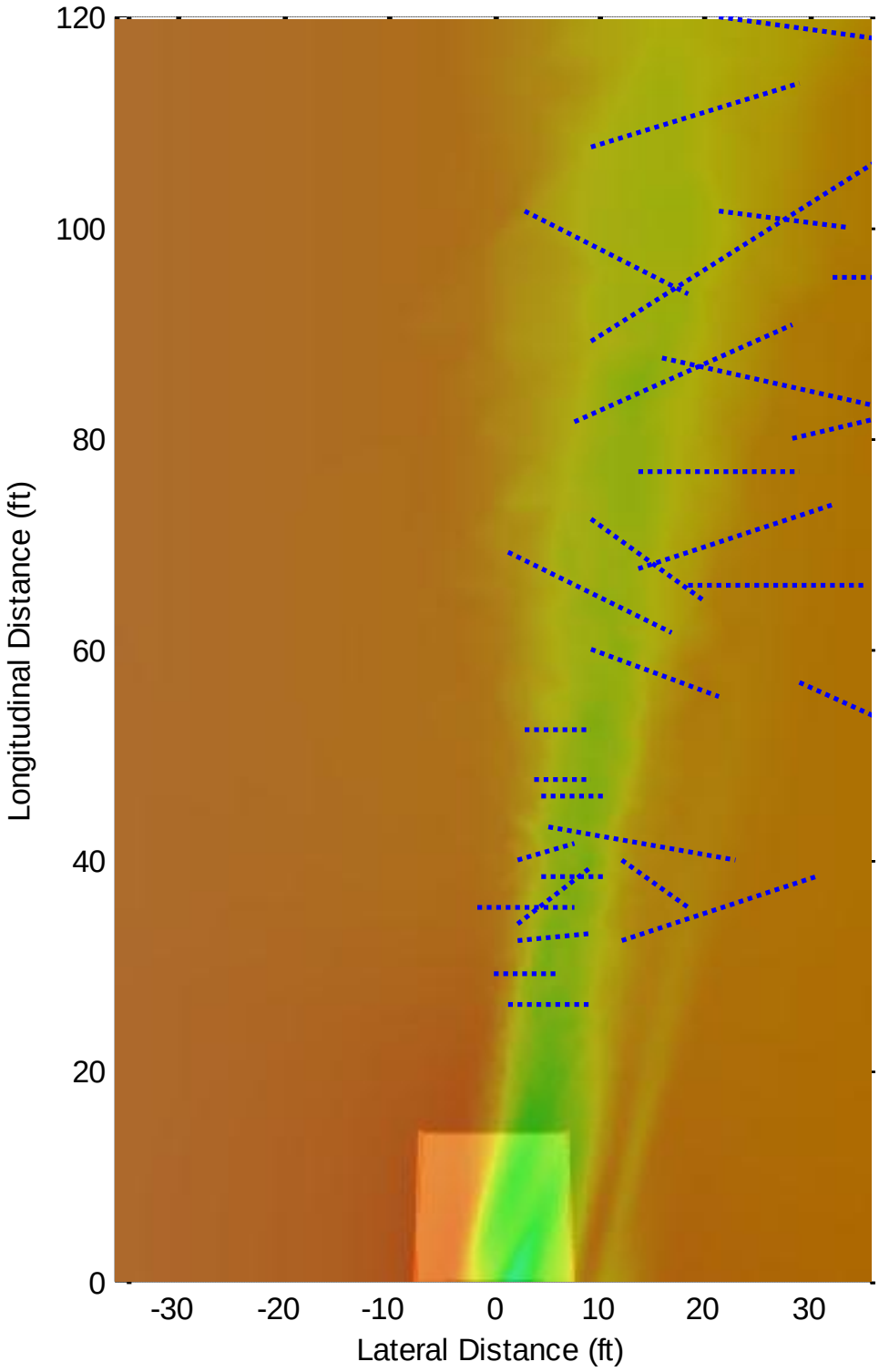

Figure 26. Measured air wake location (blue dashed lines) and CFD simulation (colored background) for β=15° at the top of the conning tower.

3.3 Top of Hangar Structure with β=30°

The second relative wind angle primarily flown during this project was at 30° off of the starboard bow. Figure 27 below displays the data acquired through flight operations compared to a CFD flow visualization of the ship's air wake for this angle at the height of the top of the hangar structure. For this wind condition, there appears to be less data that falls outside of the green area of the wake when compared to the previous two figures; however, the flight paths for this set of data rarely covered the entire width of the predicted air wake at this wind condition. The pilots noted that it was harder to refrain from controlling the helicopter during flight, often times interrupting a flight through the wake in order to get the helicopter back to the calm wind regions on either side of the wake.

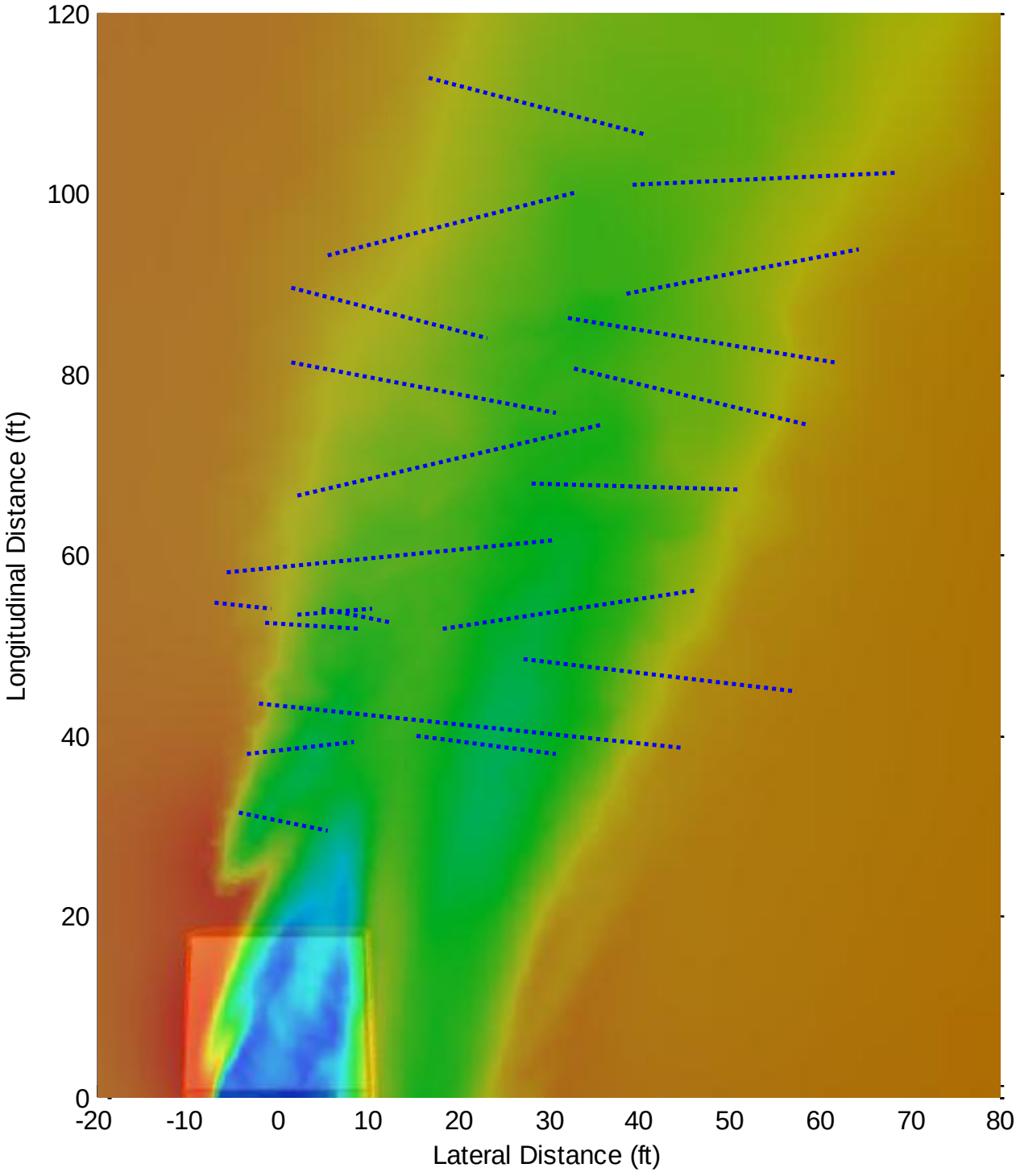

Figure 27. Measured air wake location (blue dashed lines) and CFD simulation (colored background) for β=30° at the top of the hangar structure.

3.4 Top of Conning Tower with β=30°

Figure 28 presents the data from flight operations at a relative wind angle of 30° off of the starboard bow at the height of the top of the conning tower. Many of the discussed trends from Figure 25 through Figure 27 apply to this set of data as well.

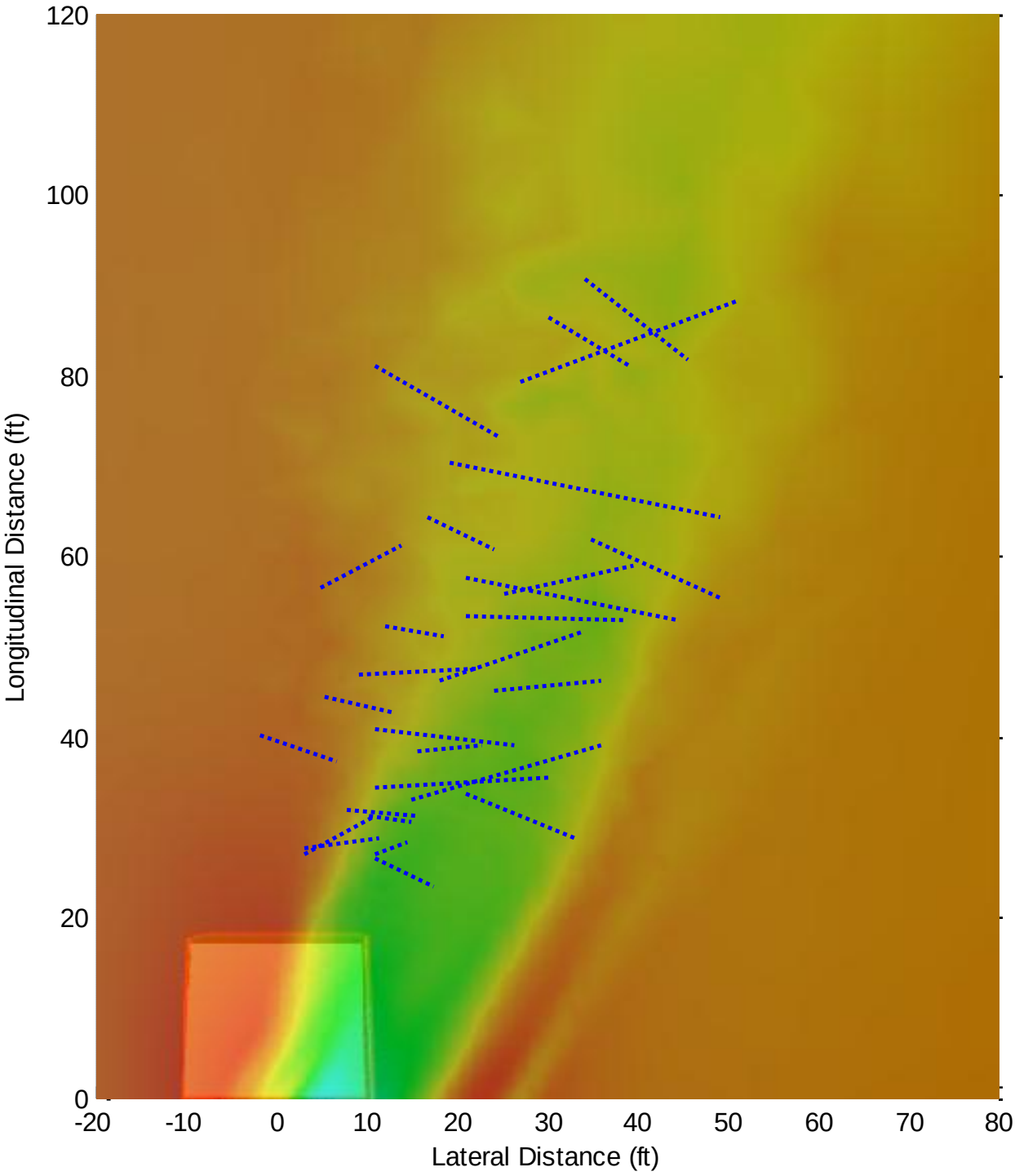

Figure 28. Measured air wake location (blue dashed lines) and CFD simulation (colored background) for β=30° at the top of the hangar structure.

4 Conclusions and Future Work

The data obtained during underway flight operations regarding the locations of the helicopter when affected by the ship's air wake corresponds well to predictions from CFD analyses for a relative wind angles of $\beta=15°$ and $\beta=30°$.

Instrumented remotely piloted helicopters can be used to map air wake interaction in regions aft of a ship. This method of investigation has been shown using an IMU, multiple GPS units, and extensive man-in-the-loop calculations and observations.

The concept of using remotely piloted helicopters to measure a ship air wake's impact on the helicopter during flight has great potential for growth and further application. Though it has been shown that separate inertial measurement units and GPS units can map the locations of the air wake's effects on the helicopter, it is a very tedious and cumbersome process that requires many hours of man-in-the-loop calculations. Automation of this data analysis process, including discrimination between pilot and ship air wake induced flight path perturbations, would be of significant value. Additionally, the helicopters are prone to mechanical failure; much care must be given to the test equipment before, during, and after each underway session.

Remotely piloted helicopters will be used in the future with the Ship Air Wake Project, but the methods of measuring the air wake's effects on the helicopter's flight may change. Being able to quantify the pilot's inputs or the reactions by heading-lock and other gyros onboard more advanced helicopters would more directly show the interaction between the air wake and the helicopter. Additionally, criteria can be developed for creating a full launch and recovery envelope for the T-Rex 600 E Super Pro onboard a YP, similar in nature to the envelope displayed in Figure 1. A study would then follow to see how the CFD simulations at different wind over deck angles correspond to the limits set by test pilots. Additionally, the ability to filter the flight data according to the actual measured relative wind angle as it shifted during flight operations would allow for the development of a method to quantify how well the measured flight data corresponds with the CFD simulations; however, with the current data, the shifting wind is not yet taken into account.

Other planned work for the Ship Air Wake Project involves modifying the geometry of the ship around the flight deck to mitigate the effects of the ship's air wake aft of the ship. A

combined CFD, wind tunnel, and *in situ* study will be performed on this topic during the 2012-2013 academic year at the U.S. Naval Academy. Additionally, continued atmospheric boundary layer testing using an array of anemometers mounted at various heights above the bow will be carried out in the summer of 2012.

References

[1] Burks, J.S., and DeProspero, F.J., "Helicopter/Aviation facility ship compatibility program SH-3/CG-11 dynamic interface evaluation," December 1975.

[2] Naval Air Systems Command, "CN NATOPS Manual," July 2009.

[3] Snyder, M.R., "Determination of ship borne helicopter launch and recovery limitations using computational fluid dynamics," 2009 Office of Naval Research Young Investigator Program.

[4] Guillot, M.J., and Walker, M.A., "Unsteady analysis of the air wake over the LPD-17," *AIAA 2000-4125: 18th Applied Aerodynamics Conference*, 14-17 August 2000.

[5] Guillot, M.J., "Computational simulation of the air wake over a naval transport vessel," *AIAA Journal*, 40(10), 2002.

[6] Lee, D., Horn, J.F., Sezer-Uzol, N., and Long, L.N, "Simulation of pilot control activity during helicopter shipboard operations," *AIAA 2003-5306: Atmospheric Flight Mechanics Conference and Exhibit*, 11-14 August 2003.

[7] Carico, D., "Rotorcraft shipboard flight test analytic options," *Proceedings 2004 IEEE Aerospace Conference*, 5(6), 13 March 2004.

[8] Lee, D., Sezer-Uzol, N., Horn, J.F., and Long, L.N., "Simulation of helicopter ship-board launch and recovery with time-accurate airwakes," *Journal of Aircraft*, 42(2), March-April 2005.

[9] Geder, J., Ramamurti, R., and Sandberg, W.C., "Ship airwake correlation analysis for the San Antonio Class Transport Dock Vessel," *Naval Research Laboratory paper MRL/MR/6410-09-9127*, 21 May 2008.

[10] Kääriä, C.H., Wang, Y., Curran, J., Forrest, J., and Owen, I. "AirDyn: An Airwake Dynamometer for measuring the impact of ship geometry on helicopter operations," 36th European Rotorcraft Forum, September 2010.

[11] Snyder, M.R., Shishkoff, J.P., Roberson, F.D., McDonald, M.C., Brownell, C.J., Luznik, L., Miklosovic, D.S., Burks, J.S., Kang, H.S., and Wilkinson, C.H. "Comparison of experimental and computational ship air wakes for YP class patrol craft," ASNE Launch & Recovery Symposium, December 2010.

[12] Metzger, J.D., Snyder, M.R., Burks, J.S., and Kang, H.S. "Measurement of Ship Air Wake Impact on a Remotely Piloted Aerial Vehicle," American Helicopter Society 68th Annual Forum, Ft. Worth, Texas, 1 May 2012.

[13] Snyder, M.R., and Kang, H.S. “Comparison of experimental and computational ship air wakes for YP class patrol craft,” AIAA 2011-7045-975: Centennial of Naval Aviation Forum, 21-22 September 2011.

[14] Pope, S.B. *Turbulent Flows*. Cambridge University Press, Cambridge, England, 2000.

[15] Watkinson, John. *Art of the Helicopter*. Elsevier, Oxford, England. 2004.

APPENDIX A: Raw Data Examples

Table 5. Example SD GPS Data Logger data output (3 seconds).

Latitude (deg)	Longitude (deg)	Altitude (m)	Heading (°β)	Speed (kts)	Year	Month	Day	Hour (GMT)	Min	Sec	mSec
38.9566	-76.4439	0.3	115.16	9.5	2012	3	2	18	58	8	0
38.9566	-76.4439	0.3	113.55	9.18	2012	3	2	18	58	8	100
38.9566	-76.4439	0.3	117.55	8.94	2012	3	2	18	58	8	300
38.9566	-76.4439	0.4	120.76	8.37	2012	3	2	18	58	8	400
38.9566	-76.4439	0.3	125.04	7.74	2012	3	2	18	58	8	500
38.9565	-76.4439	0.3	124.94	7.96	2012	3	2	18	58	8	600
38.9565	-76.4439	0.2	126.95	7.9	2012	3	2	18	58	8	700
38.9565	-76.4439	0.2	129.3	7.53	2012	3	2	18	58	8	800
38.9565	-76.4439	0.2	135.78	7.49	2012	3	2	18	58	8	900
38.9565	-76.4439	0.2	140.05	7.06	2012	3	2	18	58	9	0
38.9565	-76.4439	0.2	141.33	7.24	2012	3	2	18	58	9	100
38.9565	-76.4439	0.2	144.39	7.34	2012	3	2	18	58	9	300
38.9565	-76.4439	0.2	143.74	7.36	2012	3	2	18	58	9	400
38.9565	-76.4439	0.2	146.44	7.11	2012	3	2	18	58	9	500
38.9565	-76.4439	0.1	147.38	7.36	2012	3	2	18	58	9	600
38.9565	-76.4439	0.1	151.92	7.35	2012	3	2	18	58	9	700
38.9565	-76.4439	0.1	150.35	7.94	2012	3	2	18	58	9	800
38.9565	-76.4439	0.1	152.83	7.97	2012	3	2	18	58	9	900
38.9565	-76.4439	0.1	155.39	8.55	2012	3	2	18	58	10	0
38.9565	-76.4439	0.1	153.88	8.91	2012	3	2	18	58	10	100
38.9565	-76.4439	0.1	156.58	9.3	2012	3	2	18	58	10	300
38.9565	-76.4439	0.1	157.13	9.34	2012	3	2	18	58	10	400
38.9565	-76.4438	0.1	157.49	9.55	2012	3	2	18	58	10	500
38.9565	-76.4438	0.1	164.41	9.51	2012	3	2	18	58	10	600
38.9565	-76.4438	0.1	163.16	9.91	2012	3	2	18	58	10	700
38.9565	-76.4438	0	167.05	10.16	2012	3	2	18	58	10	800
38.9565	-76.4438	0.1	166.41	10.19	2012	3	2	18	58	10	900

Table 6. Example x-IMU data output (0.25 second).

Packet number	Gyroscope X (deg/s)	Gyroscope Y (deg/s)	Gyroscope Z (deg/s)	Accelerometer X (g)	Accelerometer Y (g)	Accelerometer Z (g)
308682	1.375	-39.875	-9.5625	1.313965	0.6591797	1.115234
308684	17.4375	-39.9375	7.5625	-1.868164	0.9492188	0.9082031
308688	-20.875	-3.8125	-31.75	-0.7929688	-0.01708984	-0.01074219
308690	-21.1875	-3	-17.1875	-1.462402	2.464844	-0.2416992
308693	-14.125	-23.375	-36.875	0.7666016	-1.880859	2.163574
308695	-15.6875	-49.5	-9.875	-0.9541016	1.873047	0.4423828
308698	-8.25	-22.6875	-29.875	-3.428223	1.92334	2.132324
308700	-36.375	-1.25	-34.6875	-0.418457	-0.237793	-0.3901367
308703	-31.4375	-2.0625	-13.0625	-0.01269531	3.253418	0.1450195
308705	-40.6875	-11.5625	-19.125	2.432617	-0.6357422	2.496094
308708	-31.5	-26.25	-9.625	-1.891602	2.878418	3.114258
308710	-23.5	-0.5	-49.75	-1.501953	0.09082031	2.652832
308713	-40.6875	11.6875	-43.625	1.058105	1.068848	1.072266
308715	-29.8125	5.625	-27.9375	0.5854492	3.582031	2.054199
308718	-33.375	-6.625	-35.625	-0.284668	-0.6743164	3.228027
308720	-27.5	-11.3125	-20.3125	-1.245605	3.203125	0.2392578
308723	-26.3125	45.875	-71.75	-1.391602	-0.237793	1.561523
308725	-26	21.4375	-27.5625	1.668457	2.116699	1.937012
308728	-29.25	13.4375	-19.375	-2.018066	1.707031	1.268066
308730	-40.4375	-12.125	-40	1.510742	-2.209473	1.361816
308733	-25.5625	-7.3125	-29	-2.805664	3.005859	1.604492
308735	-12.9375	29.25	-65.0625	1.113281	-1.946777	2.089355
308738	-11.125	7.0625	-4.625	-1.005371	2.344727	2.034668
308740	20.3125	48.5	-55.5	-0.769043	0.8564453	1.577148
308743	-12.4375	-1.5	-12.9375	-0.7495117	2.039551	2.820801
308745	15.9375	38.3125	-38.9375	-3.120605	1.358887	2.363281
308748	-11.5625	1.125	-45.0625	2.318359	-0.7402344	0.7006836
308750	-13.1875	-14.5	-34	-1.83252	3.230469	3.12207
308753	12.5625	-16.375	-43.9375	1.223633	-1.088379	2.961914
308755	-11.125	-17.3125	-25.0625	-1.907715	2.32959	2.543457
308758	23.9375	45.5625	-71.25	-1.241699	-0.1445313	2.10498
308760	19.5625	-11.5	-27.25	0.7509766	2.163086	1.706055

APPENDIX B: MATLAB Scripts

IMU Data Trim and Repackage Code (IMUunpack_V2.m)

```
%%
% ========================================== %
% Helicopter / YP IMU Unpacker               %
% Developed by: MIDN Jason D. Metzger        %
% Modified by: Hyung Suk Kang                %
% Updated: 05 February 2012                  %
% ========================================== %

% Note: Two folders are required as sub-directories from where this script
% is located - "IMU Heli" and "IMU YP". Inside each of these folders should
% be the converted .csv output files from the x-IMU GUI. Also, the data
% should be pre-trimmed due to the size of the .csv files that are being
% worked with. See Jason for further details.
clc, clear, close all, format compact, format short g
%% User Inputs
% YPFileName='00014a';
% HeliFileName='00014a';
% StartTime=[2012 2 3 12 59 17]; % YYYY MM DD hh mm ss
% EndTime  =[2012 2 3 13  7 16]; % YYYY MM DD hh mm ss
% OutputFileName='Flight1';

YPFileName='00014f';
HeliFileName='00014f';
StartTime=[2012 2 3 14 49 2]; % YYYY MM DD hh mm ss
EndTime=[2012 2 3 14 56 32]; % YYYY MM DD hh mm ss
SampleRate=128; % IMU Data Sample Rate [Hz]
OutputFileName='Flight6';


%% Define parameter
SampleRate=128; % IMU Data Sample Rate [Hz]
dt=1/SampleRate;
g_o=9.80665;


%% Data Import and Trim for DataTime
%Heli.DateTime(1,:) - packet #
H=waitbar(0/12,'Importing and Trimming Data...Please Wait');
Heli.DateTime=xlsread(['IMU Heli/',HeliFileName,'_DateTime.csv']);
waitbar(1/12,H);

% Find the start and end packet # and line #
for i=1:length(Heli.DateTime)     %data line
    if (Heli.DateTime(i,2:7)==StartTime)
        kSTh=Heli.DateTime(i,1);  %Start packet number at final edge
        nl_DT_start=i;            %start line number
    end
    if (Heli.DateTime(i,2:7)==EndTime)
```

```
        kETh=Heli.DateTime(i,1);  %End packet number at final edge
        nl_DT_end=i;              %end line number
    end
end
Heli.DateTime=Heli.DateTime(nl_DT_start:nl_DT_end,:);
waitbar(2/12,H);

%% Data Import and Trim for EulerAngles
Heli.Euler=xlsread(['IMU Heli/',HeliFileName,'_EulerAngles.csv']);
waitbar(3/12,H);

% Find the start line number, kEAsh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (Heli.Euler(i,1)>=kSTh)
        iflag=1;
        kEAsh=i;
    end
end

% Find the end line number, kEAeh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (Heli.Euler(i,1)>=kETh)
        iflag=1;
        kEAeh=i;
    end
end
Heli.Euler=Heli.Euler(kEAsh:kEAeh,:);
waitbar(4/12,H);

%% Data Import and Trim for IMU Heli
Heli.CalIntMag=xlsread(['IMU Heli/',HeliFileName,'_CalInertialAndMag.csv']);
waitbar(5/12,H);

% Find the start line number, kCIMsh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (Heli.CalIntMag(i,1)>=kSTh)
        iflag=1;
        kCIMsh=i;
    end
end

% Find the end line number, kCIMeh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (Heli.CalIntMag(i,1)>=kETh)
```

```
        iflag=1;
        kCIMeh=i;
    end
end
Heli.CalIntMag=Heli.CalIntMag(kCIMsh:kCIMeh,:);
waitbar(6/12,H);




%% YP IMU Data
%% Data Import and Trim for DataTime
%Heli.DateTime(1,:) - packet #
YP.DateTime=xlsread(['IMU YP/',YPFileName,'_DateTime.csv']);
waitbar(7/12,H);

% Find the start and end packet # and line #
for i=1:length(YP.DateTime)     %data line
    if (YP.DateTime(i,2:7)==StartTime)
        kSTh=YP.DateTime(i,1);  %Start packet number at final edge
        nl_DT_start=i;            %start line number
    end
    if (YP.DateTime(i,2:7)==EndTime)
        kETh=YP.DateTime(i,1);  %End packet number at final edge
        nl_DT_end=i;              %end line number
    end
end
YP.DateTime=YP.DateTime(nl_DT_start:nl_DT_end,:);
waitbar(8/12,H);

%% Data Import and Trim for EulerAngles
YP.Euler=xlsread(['IMU YP/',YPFileName,'_EulerAngles.csv']);
waitbar(9/12,H);

% Find the start line number, kEAsh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (YP.Euler(i,1)>=kSTh)
        iflag=1;
        kEAsh=i;
    end
end

% Find the end line number, kEAeh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (YP.Euler(i,1)>=kETh)
        iflag=1;
        kEAeh=i;
    end
end
```

```
YP.Euler=YP.Euler(kEAsh:kEAeh,:);
waitbar(10/12,H);

%% Data Import and Trim for IMU Heli
YP.CalIntMag=xlsread(['IMU YP/',YPFileName,'_CalInertialAndMag.csv']);
waitbar(11/12,H);

% Find the start line number, kCIMsh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (YP.CalIntMag(i,1)>=kSTh)
        iflag=1;
        kCIMsh=i;
    end
end

% Find the end line number, kCIMeh
i=0;
iflag=0;
while (iflag==0)
    i=i+1;
    if (YP.CalIntMag(i,1)>=kETh)
        iflag=1;
        kCIMeh=i;
    end
end
YP.CalIntMag=YP.CalIntMag(kCIMsh:kCIMeh,:);
waitbar(12/12,H);


%% Close and Save
close(H)
save(OutputFileName,'Heli','YP','g_o','dt')
```

Raw Data Video Creator (HeliMovieMaker_V2.m)

```
%%
% ========================================== %
% Helicopter Data Movie Maker                %
% Developed by: MIDN Jason D. Metzger        %
% Modified by: Hyung Suk Kang                %
% Updated: 06 February 2012                  %
% ========================================== %

% Note: This script extracts data from a "Flight#.mat" file and plots the
% accelerometer, euler angles, and gyroscope data as a function of time and
% saves the plots as a movie file.
clc, clear, close all, format compact, format short g
%% User Inputs
% load Flight1;         % Load the Flight#.mat file you want
% OutputName='Flight1'; % Base name of the output .avi files
load Flight1; % Load the Flight#.mat file you want
```

```
OutputName='Flight1'; % Base name of the output .avi files

%% Parameters
Tmovie=60;    % Time length for each movie [sec]
Tspan =10;    % Time span in Time-axis [sec]
freq  =128;   % 128 Hz for data & 64 Hz for Time
incre =10;    % data increment to update movie

dt=1./freq;
ndata_movie=freq*Tmovie; % # data per movie
nmovie=fix( length(Heli.CalIntMag) / (freq*Tmovie) ) + 1; % # movies
nframe=(freq*Tmovie)/incre;   % # frames per each movie (1024)
xtime=-Tspan:dt:0;   % 10 sec length
xtime=xtime';

%% Pre-allocation:
Xaccel=zeros(Tspan*freq+1,1);   % for Tspan + 1
Yaccel=zeros(Tspan*freq+1,1);
Zaccel=zeros(Tspan*freq+1,1);
Xgyro =zeros(Tspan*freq+1,1);   % for Tspan + 1
Ygyro =zeros(Tspan*freq+1,1);
Zgyro =zeros(Tspan*freq+1,1);

%% Min and Max
maxXaccel=max(Heli.CalIntMag(:,5));   % g's
minXaccel=min(Heli.CalIntMag(:,5));
maxYaccel=max(Heli.CalIntMag(:,6));
minYaccel=min(Heli.CalIntMag(:,6));
maxZaccel=max(Heli.CalIntMag(:,7));
minZaccel=min(Heli.CalIntMag(:,7));
maxXgyro =max(Heli.CalIntMag(:,2));   % deg/s
minXgyro =min(Heli.CalIntMag(:,2));
maxYgyro =max(Heli.CalIntMag(:,3));
minYgyro =min(Heli.CalIntMag(:,3));
maxZgyro =max(Heli.CalIntMag(:,4));
minZgyro =min(Heli.CalIntMag(:,4));

%% Data Extraction and Plotting for Gyro data at Heli
for imovie=1:nmovie
    if (imovie==nmovie) % Find nframe for the last movie clip
        nleft=mod(length(Heli.CalIntMag),(freq*Tmovie));
        nframe=fix(nleft/incre);  % Cut-off small last portion.
    end

    hfig=figure(1);
    for k=1:nframe
%        timer =fix((k*incre+1)/2);
        itime =fix( ((imovie-1)*ndata_movie + k*incre + 1) / 2 );
        hour  =num2str(Heli.DateTime(itime,5));
        minute=num2str(Heli.DateTime(itime,6));
        second=num2str(Heli.DateTime(itime,7));

        is=(imovie-1)*ndata_movie + (k-1)*incre + 1; % start point
        ie=is + incre;                               % end point
        if (is<Tspan*freq+1) % Initial time: fill with 0.
```

```
            Xdata=Heli.CalIntMag(1:ie,2);
            Ydata=Heli.CalIntMag(1:ie,3);
            Zdata=Heli.CalIntMag(1:ie,4);
            Xgyro=vertcat(zeros(length(xtime)-length(Xdata),1),Xdata);
            Ygyro=vertcat(zeros(length(xtime)-length(Ydata),1),Ydata);
            Zgyro=vertcat( ones(length(xtime)-length(Zdata),1),Zdata);
        elseif ( (is>=Tspan*freq+1) && (ie<=length(Heli.CalIntMag)) )
            Xgyro=Heli.CalIntMag(ie-(length(xtime)-1):ie,2);
            Ygyro=Heli.CalIntMag(ie-(length(xtime)-1):ie,3);
            Zgyro=Heli.CalIntMag(ie-(length(xtime)-1):ie,4);
        else
            disp('Ckeck data length or arrays');
        end

        suptitle(['Gyroscope Data at ',hour,':',minute,':',second])
        subplot(3,1,1)
        plot(xtime,Xgyro,'b-')
        axis([-Tspan 0 -100 100]);
        ylabel('X (deg/sec)')
        subplot(3,1,2)
        plot(xtime,Ygyro,'r-')
        axis([-Tspan 0 -100 100]);
        ylabel('Y (deg/sec)')
        subplot(3,1,3)
        plot(xtime,Zgyro,'g-')
        axis([-Tspan 0 -100 100]);
        ylabel('Z (deg/sec)')
        xlabel('Time (sec)')
        Mgyro(k)=getframe(hfig);
    end

    close(1);
    movie2avi(Mgyro,['Data
Videos\',OutputName,'_gyro_Heli',num2str(imovie),'.avi'],'compression','none'
);
    clear Mgyro

end   % End of gyro at Heli


%% Data Extraction and Plotting for Acceration at Heli
for imovie=1:nmovie
    if (imovie==nmovie) % Find nframe for the last movie clip
        nleft=mod(length(Heli.CalIntMag),(freq*Tmovie));
        nframe=fix(nleft/incre);  % Cut-off small last portion.
    end

    hfig=figure(1);
    for k=1:nframe
%         timer =fix((k*incre+1)/2);
        itime =fix( ((imovie-1)*ndata_movie + k*incre + 1) / 2 );
        hour  =num2str(Heli.DateTime(itime,5));
        minute=num2str(Heli.DateTime(itime,6));
        second=num2str(Heli.DateTime(itime,7));
```

```
        is=(imovie-1)*ndata_movie + (k-1)*incre + 1; % start point
        ie=is + incre;                               % end point
        if (is<Tspan*freq+1) % Initial time: fill with 0.
            Xdata=Heli.CalIntMag(1:ie,5);
            Ydata=Heli.CalIntMag(1:ie,6);
            Zdata=Heli.CalIntMag(1:ie,7);
            Xaccel=vertcat(zeros(length(xtime)-length(Xdata),1),Xdata);
            Yaccel=vertcat(zeros(length(xtime)-length(Ydata),1),Ydata);
            Zaccel=vertcat( ones(length(xtime)-length(Zdata),1),Zdata);
        elseif ( (is>=Tspan*freq+1) && (ie<=length(Heli.CalIntMag)) )
            Xaccel=Heli.CalIntMag(ie-(length(xtime)-1):ie,5);
            Yaccel=Heli.CalIntMag(ie-(length(xtime)-1):ie,6);
            Zaccel=Heli.CalIntMag(ie-(length(xtime)-1):ie,7);
        else
            disp('Ckeck data length or arrays');
        end

        suptitle(['Accelerometer Data at ',hour,':',minute,':',second])
        subplot(3,1,1)
        plot(xtime,Xaccel,'b-')
        axis([-Tspan 0 -10 10]);
        ylabel('X (g)')
        subplot(3,1,2)
        plot(xtime,Yaccel,'r-')
        axis([-Tspan 0 -10 10]);
        ylabel('Y (g)')
        subplot(3,1,3)
        plot(xtime,Zaccel,'g-')
        axis([-Tspan 0 -10 10]);
        ylabel('Z (g)')
        xlabel('Time (sec)')
        Maccel(k)=getframe(hfig);
    end

    close(1);
    movie2avi(Maccel,['Data
Videos\',OutputName,'_accel_Heli',num2str(imovie),'.avi'],'compression','none
');
    clear Maccel

end    % End of Accel


%% Data Extraction and Plotting for Gyro data at YP
for imovie=1:nmovie
    if (imovie==nmovie) % Find nframe for the last movie clip
        nleft=mod(length(YP.CalIntMag),(freq*Tmovie));
        nframe=fix(nleft/incre);  % Cut-off small last portion.
    end

    hfig=figure(1);
    for k=1:nframe
%        timer =fix((k*incre+1)/2);
        itime =fix( ((imovie-1)*ndata_movie + k*incre + 1) / 2 );
        hour  =num2str(YP.DateTime(itime,5));
        minute=num2str(YP.DateTime(itime,6));
```

```
        second=num2str(YP.DateTime(itime,7));

        is=(imovie-1)*ndata_movie + (k-1)*incre + 1; % start point
        ie=is + incre;                               % end point
        if (is<Tspan*freq+1) % Initial time: fill with 0.
            Xdata=YP.CalIntMag(1:ie,2);
            Ydata=YP.CalIntMag(1:ie,3);
            Zdata=YP.CalIntMag(1:ie,4);
            Xgyro=vertcat(zeros(length(xtime)-length(Xdata),1),Xdata);
            Ygyro=vertcat(zeros(length(xtime)-length(Ydata),1),Ydata);
            Zgyro=vertcat( ones(length(xtime)-length(Zdata),1),Zdata);
        elseif ( (is>=Tspan*freq+1) && (ie<=length(YP.CalIntMag)) )
            Xgyro=YP.CalIntMag(ie-(length(xtime)-1):ie,2);
            Ygyro=YP.CalIntMag(ie-(length(xtime)-1):ie,3);
            Zgyro=YP.CalIntMag(ie-(length(xtime)-1):ie,4);
        else
            disp('Ckeck data length or arrays');
        end

        suptitle(['Gyroscope Data at ',hour,':',minute,':',second])
        subplot(3,1,1)
        plot(xtime,Xgyro,'b-')
        axis([-Tspan 0 -4 -2]);
        ylabel('X (deg/sec)')
        subplot(3,1,2)
        plot(xtime,Ygyro,'r-')
        axis([-Tspan 0 -2 0]);
        ylabel('Y (deg/sec)')
        subplot(3,1,3)
        plot(xtime,Zgyro,'g-')
        axis([-Tspan 0 -2 0]);
        ylabel('Z (deg/sec)')
        xlabel('Time (sec)')
        Mgyro(k)=getframe(hfig);
    end

    close(1);
    movie2avi(Mgyro,['Data
Videos\',OutputName,'_gyro_YP',num2str(imovie),'.avi'],'compression','none');
    clear Mgyro

end   % End of gyro at YP


%% Data Extraction and Plotting for Acceration at YP
for imovie=1:nmovie
    if (imovie==nmovie) % Find nframe for the last movie clip
        nleft=mod(length(YP.CalIntMag),(freq*Tmovie));
        nframe=fix(nleft/incre);  % Cut-off small last portion.
    end

    hfig=figure(1);
    for k=1:nframe
%         timer =fix((k*incre+1)/2);
        itime =fix( ((imovie-1)*ndata_movie + k*incre + 1) / 2 );
```

```
        hour  =num2str(YP.DateTime(itime,5));
        minute=num2str(YP.DateTime(itime,6));
        second=num2str(YP.DateTime(itime,7));

        is=(imovie-1)*ndata_movie + (k-1)*incre + 1; % start point
        ie=is + incre;                               % end point
        if (is<Tspan*freq+1) % Initial time: fill with 0.
            Xdata=YP.CalIntMag(1:ie,5);
            Ydata=YP.CalIntMag(1:ie,6);
            Zdata=YP.CalIntMag(1:ie,7);
            Xaccel=vertcat(zeros(length(xtime)-length(Xdata),1),Xdata);
            Yaccel=vertcat(zeros(length(xtime)-length(Ydata),1),Ydata);
            Zaccel=vertcat( ones(length(xtime)-length(Zdata),1),Zdata);
        elseif ( (is>=Tspan*freq+1) && (ie<=length(YP.CalIntMag)) )
            Xaccel=YP.CalIntMag(ie-(length(xtime)-1):ie,5);
            Yaccel=YP.CalIntMag(ie-(length(xtime)-1):ie,6);
            Zaccel=YP.CalIntMag(ie-(length(xtime)-1):ie,7);
        else
            disp('Ckeck data length or arrays');
        end

        suptitle(['Accelerometer Data at ',hour,':',minute,':',second])
        subplot(3,1,1)
        plot(xtime,Xaccel,'b-')
        axis([-Tspan 0 -0.1 0.1]);
        ylabel('X (g)')
        subplot(3,1,2)
        plot(xtime,Yaccel,'r-')
        axis([-Tspan 0 -0.1 0.1]);
        ylabel('Y (g)')
        subplot(3,1,3)
        plot(xtime,Zaccel,'g-')
        axis([-Tspan 0 1.0 1.2]);
        ylabel('Z (g)')
        xlabel('Time (sec)')
        Maccel(k)=getframe(hfig);
    end

    close(1);
    movie2avi(Maccel,['Data
Videos\',OutputName,'_accel_YP',num2str(imovie),'.avi'],'compression','none')
;
    clear Maccel

end    % End of Accel
```

Filtered Data Movie Creator (IMU_statistics.m)

```
%%
% ========================================== %
% Statistics for IMU data                    %
% Developed by: Hyung Suk Kang               %
% Modified by: MIDN Jason D. Metzger         %
% Updated: 15 February 2012                  %
% ========================================== %

% Note: This script extracts data from a "Flight#.mat" file and plots the
% accelerometer, euler angles, and gyroscope data as a function of time and
% saves the plots as a movie file.
clc, clear, close all, format compact, format short g
%% User Inputs
load Flight4;         % Load the Flight#.mat file you want
OutputName='Flight04'; % Base name of the output .avi files

%% Parameters
ndim=3;       % # dimension
freq  =128;   % 128 Hz for data & 64 Hz for Time
nfilter=16;   % Filter width

dt=1./freq;
ndata=length(Heli.CalIntMag);   % # data


%% Change variables
agyro(:,1)=Heli.CalIntMag(:,2);   % Angle rates
agyro(:,2)=Heli.CalIntMag(:,3);
agyro(:,3)=Heli.CalIntMag(:,4);
accel(:,1)=Heli.CalIntMag(:,5);   % Accelerations
accel(:,2)=Heli.CalIntMag(:,6);
accel(:,3)=Heli.CalIntMag(:,7);


%% Statistics
[meangx,meangy,meangz,rmsgx,rmsgy,rmsgz,guv,gvw,gwu,gke,Sgx,Sgy,Sgz,Fgx,Fgy,F
gz]=...
     moment(ndata,1,agyro(:,1),agyro(:,2),agyro(:,3));
[meanax,meanay,meanaz,rmsax,rmsay,rmsaz,auv,avw,awu,ake,Sax,Say,Saz,Fax,Fay,F
az]=...
     moment(ndata,1,accel(:,1),accel(:,2),accel(:,3));
Limit_gx = 1.*rmsgx;

% Fluctuations of the data
agyro(:,1)=agyro(:,1)-meangx;
agyro(:,2)=agyro(:,2)-meangy;
agyro(:,3)=agyro(:,3)-meangz;
accel(:,1)=accel(:,1)-meanax;
accel(:,2)=accel(:,2)-meanay;
accel(:,3)=accel(:,3)-meanaz;

% Filtered data of the fluctuation
[fagyro(:,1)]=Filtered_field_Box(agyro(:,1),nfilter);
```

```
[fagyro(:,2)]=Filtered_field_Box(agyro(:,2),nfilter);
[fagyro(:,3)]=Filtered_field_Box(agyro(:,3),nfilter);
[faccel(:,1)]=Filtered_field_Box(accel(:,1),nfilter);
[faccel(:,2)]=Filtered_field_Box(accel(:,2),nfilter);
[faccel(:,3)]=Filtered_field_Box(accel(:,3),nfilter);



[urms]=Partial_stat_filtered(fagyro(:,1),2*freq);


% Try to capture severe motions
i=0;
for k=1:ndata
%     if (urms(k)>=Limit_gx)
    if (urms(k)>=12.)
        i=i+1;
        itime=fix((k+1)/2);
        Event.Time(i,1)=Heli.DateTime(itime,5);
        Event.Time(i,2)=Heli.DateTime(itime,6);
        Event.Time(i,3)=Heli.DateTime(itime,7);
    end

end



%% Parameters
Tmovie=60;    % Time length for each movie [sec]
Tspan =10;    % Time span in Time-axis [sec]
freq  =128;   % 128 Hz for data & 64 Hz for Time
incre =10;    % data increment to update movie

dt=1./freq;
ndata_movie=freq*Tmovie; % # data per movie
nmovie=fix( length(Heli.CalIntMag) / (freq*Tmovie) ) + 1; % # movies
nframe=(freq*Tmovie)/incre;   % # frames per each movie (1024)
xtime=-Tspan:dt:0;   % 10 sec length
xtime=xtime';

%% Pre-allocation:
Xaccel=zeros(Tspan*freq+1,1);   % for Tspan + 1
Yaccel=zeros(Tspan*freq+1,1);
Zaccel=zeros(Tspan*freq+1,1);
Xgyro =zeros(Tspan*freq+1,1);   % for Tspan + 1
Ygyro =zeros(Tspan*freq+1,1);
Zgyro =zeros(Tspan*freq+1,1);


%% Data Extraction and Plotting for Gyro data.
for imovie=1:nmovie
    if (imovie==nmovie) % Find nframe for the last movie clip
        nleft=mod(length(Heli.CalIntMag),(freq*Tmovie));
        nframe=fix(nleft/incre);  % Cut-off small last portion.
```

```
    end

    hfig=figure(1);
    for k=1:nframe
%         timer =fix((k*incre+1)/2);
        itime =fix( ((imovie-1)*ndata_movie + k*incre + 1) / 2 );
        hour  =num2str(Heli.DateTime(itime,5));
        minute=num2str(Heli.DateTime(itime,6));
        second=num2str(Heli.DateTime(itime,7));

        is=(imovie-1)*ndata_movie + (k-1)*incre + 1; % start point
        ie=is + incre;                               % end point
        if (is<Tspan*freq+1) % Initial time: fill with 0.
            Xdata=fagyro(1:ie,1);
            Ydata=fagyro(1:ie,2);
            Zdata=fagyro(1:ie,3);
            fXgyro=vertcat(zeros(length(xtime)-length(Xdata),1),Xdata);
            fYgyro=vertcat(zeros(length(xtime)-length(Ydata),1),Ydata);
            fZgyro=vertcat( ones(length(xtime)-length(Zdata),1),Zdata);
        elseif ( (is>=Tspan*freq+1) && (ie<=length(Heli.CalIntMag)) )
            Xgyro=Heli.CalIntMag(ie-(length(xtime)-1):ie,2);
            Ygyro=Heli.CalIntMag(ie-(length(xtime)-1):ie,3);
            Zgyro=Heli.CalIntMag(ie-(length(xtime)-1):ie,4);
            fXgyro=fagyro(ie-(length(xtime)-1):ie,1);
            fYgyro=fagyro(ie-(length(xtime)-1):ie,2);
            fZgyro=fagyro(ie-(length(xtime)-1):ie,3);
        else
            disp('Check data length or arrays');
        end
        hold on
        suptitle(['Gyroscope FData at ',hour,':',minute,':',second])
%         subplot(3,1,1)
%         plot(xtime,Xgyro,'Color',[.7 .7 .7])
%         hold on
%         plot(xtime,fXgyro,'r-','Linewidth',1.5)
%         axis([-Tspan 0 -100 100]);
%         ylabel('X (deg/sec)')
%         hold on
%         subplot(3,1,2)
%         plot(xtime,Ygyro,'Color',[.7 .7 .7])
%         hold on
%         plot(xtime,fYgyro,'r-','Linewidth',1.5)
%         axis([-Tspan 0 -100 100]);
%         ylabel('Y (deg/sec)')
%         hold on
%         subplot(3,1,3)
%         plot(xtime,Zgyro,'Color',[.7 .7 .7])
%         hold on
%         plot(xtime,fZgyro,'r-','Linewidth',1.5)
%         axis([-Tspan 0 -100 100]);
%         ylabel('Z (deg/sec)')
%         xlabel('Time (sec)')

        subplot(2,1,1)
        plot(xtime,Xgyro,'Color',[.7 .7 .7])
        hold on
```

```
        plot(xtime,fXgyro,'r-','Linewidth',1.5)
        axis([-Tspan 0 -100 100]);
        ylabel('X (deg/sec)')
        hold on
        subplot(2,1,2)
        plot(xtime,Ygyro,'Color',[.7 .7 .7])
        hold on
        plot(xtime,fYgyro,'r-','Linewidth',1.5)
        axis([-Tspan 0 -100 100]);
        ylabel('Y (deg/sec)')
        xlabel('Time (sec)')
        Mgyro(k)=getframe(hfig);

    end

    close(1);
    movie2avi(Mgyro,['Data
Videos\',OutputName,'_fg',num2str(imovie),'.avi'],'compression','none');
    clear Mgyro

end   % End of gyro

return

%% Data Extraction and Plotting for Acceration.
for imovie=1:nmovie
    if (imovie==nmovie) % Find nframe for the last movie clip
        nleft=mod(length(Heli.CalIntMag),(freq*Tmovie));
        nframe=fix(nleft/incre);  % Cut-off small last portion.
    end

    hfig=figure(1);
    for k=1:nframe
%         timer =fix((k*incre+1)/2);
        itime =fix( ((imovie-1)*ndata_movie + k*incre + 1) / 2 );
        hour  =num2str(Heli.DateTime(itime,5));
        minute=num2str(Heli.DateTime(itime,6));
        second=num2str(Heli.DateTime(itime,7));

        is=(imovie-1)*ndata_movie + (k-1)*incre + 1; % start point
        ie=is + incre;                               % end point
        if (is<Tspan*freq+1) % Initial time: fill with 0.
            Xdata=faccel(1:ie,1);
            Ydata=faccel(1:ie,2);
            Zdata=faccel(1:ie,3);
            fXaccel=vertcat(zeros(length(xtime)-length(Xdata),1),Xdata);
            fYaccel=vertcat(zeros(length(xtime)-length(Ydata),1),Ydata);
            fZaccel=vertcat( ones(length(xtime)-length(Zdata),1),Zdata);
        elseif ( (is>=Tspan*freq+1) && (ie<=length(Heli.CalIntMag)) )
            Xaccel=Heli.CalIntMag(ie-(length(xtime)-1):ie,5);
            Yaccel=Heli.CalIntMag(ie-(length(xtime)-1):ie,6);
            Zaccel=Heli.CalIntMag(ie-(length(xtime)-1):ie,7);
            fXaccel=faccel(ie-(length(xtime)-1):ie,1);
            fYaccel=faccel(ie-(length(xtime)-1):ie,2);
            fZaccel=faccel(ie-(length(xtime)-1):ie,3);
```

```
        else
            disp('Ckeck data length or arrays');
        end

        suptitle(['Accelerometer FData at ',hour,':',minute,':',second])
        subplot(3,1,1)
        plot(xtime,Xaccel,'y-')
        hold;
        plot(xtime,fXaccel,'b-')
        axis([-Tspan 0 -1 1]);
        ylabel('X (g)')

        subplot(3,1,2)
        plot(xtime,Yaccel,'y-')
        hold;
        plot(xtime,fYaccel,'r-')
        axis([-Tspan 0 -1 1]);
        ylabel('Y (g)')

        subplot(3,1,3)
        plot(xtime,Zaccel,'y-')
        hold;
        plot(xtime,fZaccel,'g-')
        axis([-Tspan 0 -1 1]);
        ylabel('Z (g)')
        xlabel('Time (sec)')
        Maccel(k)=getframe(hfig);
    end

    close(1);
    movie2avi(Maccel,['Data
Videos\',OutputName,'_fa',num2str(imovie),'.avi'],'compression','none');
    clear Maccel

end    % End of Accel
```

Relative Position Calculator from GPS Data (GPScrunch2.m)

```
%%
% ================================== %
% GPS Data Cruncher (Dual Version)   %
% Developed by: MIDN Jason D. Metzger %
% Updated: 14 March 2012             %
% ================================== %

% Note: This script is to be used when separate GPS data sources used. The
% GPS units do not necessarily need to be the same...attention must be
% paid, though, to how each data logger organizes the data. The script may
% have to be slightly modified to reflect different units used.
clc, clear, format compact, close all
%% User Inputs
load Flight2 % input / output flight data file
OutputFileName='Flight2'; % should match the previous line
```

```
HeliGPSName='GPSXX023'; % root name of GPS .csv file for helicopter
YPGPSName='12030206'; % root name of GPS .csv file for YP
StartTime=[2012 3 2 14 11 54]; % manually add ~5sec buffer
EndTime=[2012 3 2 14 19 16]; % manually add ~5sec buffer
%% Data Import and Trim
rawHeli=xlsread(['GPS Heli/',HeliGPSName,'.csv']);
rawYP=xlsread(['GPS YP/',YPGPSName,'.csv']);

for nn=1:length(rawHeli)
    rawHeli(nn,14)=rawHeli(nn,14)-5; % convert GMT to EST [hrs]
    if rawHeli(nn,11:16)==StartTime
        kSTh=nn;
    elseif rawHeli(nn,11:16)==EndTime
        kETh=nn;
    end
end

StartTimemod=10000*StartTime(4)+100*StartTime(5)+StartTime(6);
EndTimemod=10000*EndTime(4)+100*EndTime(5)+EndTime(6);

for nn=1:length(rawYP)
    rawYP(nn,4)=rawYP(nn,4)-50000; % convert GMT to EST [hrs]
    if rawYP(nn,4)==StartTimemod
        kSTy=nn;
    elseif rawYP(nn,4)==EndTimemod
        kETy=nn;
    end
end

GPS.h.rawdata.latitude=rawHeli(kSTh:kETh,1); % [deg]
GPS.h.rawdata.longitude=rawHeli(kSTh:kETh,2); % [deg]
GPS.h.rawdata.altitude=rawHeli(kSTh:kETh,3); % [m?]
GPS.h.rawdata.heading=rawHeli(kSTh:kETh,4); % beta [deg]
GPS.h.rawdata.speed=rawHeli(kSTh:kETh,5); % [kts]
GPS.h.rawdata.hour=rawHeli(kSTh:kETh,14);
GPS.h.rawdata.min=rawHeli(kSTh:kETh,15);
GPS.h.rawdata.sec=rawHeli(kSTh:kETh,16);
GPS.h.rawdata.msec=rawHeli(kSTh:kETh,17);

GPS.y.rawdata.latitude=rawYP(kSTy:kETy,5); % [deg]
GPS.y.rawdata.longitude=rawYP(kSTy:kETy,6); % [deg]
GPS.y.rawdata.altitude=rawYP(kSTy:kETy,7); % [m?]
GPS.y.rawdata.heading=rawYP(kSTy:kETy,9); % beta [deg]
GPS.y.rawdata.speed=rawYP(kSTy:kETy,8); % [kts?]
GPS.y.rawdata.hour=fix(rawYP(kSTy:kETy,4)/10000);
GPS.y.rawdata.min=fix((rawYP(kSTy:kETy,4)-GPS.y.rawdata.hour*10000)/100);
GPS.y.rawdata.sec=(rawYP(kSTy:kETy,4)-GPS.y.rawdata.hour*10000-
GPS.y.rawdata.min*100);
GPS.y.rawdata.msec=zeros(length(GPS.y.rawdata.speed),1);

%% Fill in missing data on a 10Hz standard
kk=0; nn=0;
hour=StartTime(4);
min=StartTime(5)-1;
sec=StartTime(6);
msec=900;
```

```
while kk==0
    nn=nn+1;
    msec=msec+100;
    if msec>=1000
        msec=0;
        sec=sec+1;
    end
    if sec>=60
        sec=0;
        min=min+1;
    end
    if min>=60
        min=0;
        hour=hour+1;
    end
    if hour>=24
        hour=0;
    end
    GPStime(nn,:)=[hour min sec msec];
    if GPStime(nn,1:3)==EndTime(4:6)
        kk=1;
    end
end

GPS.h.latitude=zeros(length(GPStime),1);
GPS.h.longitude=zeros(length(GPStime),1);
GPS.h.altitude=zeros(length(GPStime),1);
GPS.h.heading=zeros(length(GPStime),1);
GPS.h.speed=zeros(length(GPStime),1);
GPS.h.hour=GPStime(:,1);
GPS.h.min=GPStime(:,2);
GPS.h.sec=GPStime(:,3);
GPS.h.msec=GPStime(:,4);

GPS.y.latitude=zeros(length(GPStime),1);
GPS.y.longitude=zeros(length(GPStime),1);
GPS.y.altitude=zeros(length(GPStime),1);
GPS.y.heading=zeros(length(GPStime),1);
GPS.y.speed=zeros(length(GPStime),1);
GPS.y.hour=GPStime(:,1);
GPS.y.min=GPStime(:,2);
GPS.y.sec=GPStime(:,3);
GPS.y.msec=GPStime(:,4);

kkh=1;
kky=1;

for kk=2:length(GPStime)
    if GPS.h.hour(kk)==GPS.h.rawdata.hour(kkh) &&
GPS.h.min(kk)==GPS.h.rawdata.min(kkh)...
            && GPS.h.sec(kk)==GPS.h.rawdata.sec(kkh) &&
GPS.h.msec(kk)==GPS.h.rawdata.msec(kkh)
        GPS.h.latitude(kk)=GPS.h.rawdata.latitude(kkh);
        GPS.h.longitude(kk)=GPS.h.rawdata.longitude(kkh);
        GPS.h.altitude(kk)=GPS.h.rawdata.altitude(kkh);
        GPS.h.heading(kk)=GPS.h.rawdata.heading(kkh);
```

```
        GPS.h.speed(kk)=GPS.h.rawdata.speed(kkh);
        kkh=kkh+1;
    else
        GPS.h.latitude(kk)=GPS.h.latitude(kk-1);
        GPS.h.longtiude(kk)=GPS.h.longitude(kk-1);
        GPS.h.altitude(kk)=GPS.h.altitude(kk-1);
        GPS.h.heading(kk)=GPS.h.heading(kk-1);
        GPS.h.speed(kk)=GPS.h.speed(kk-1);
    end

    if GPS.y.hour(kk)==GPS.y.rawdata.hour(kky) &&
GPS.y.min(kk)==GPS.y.rawdata.min(kky)...
            && GPS.y.sec(kk)==GPS.y.rawdata.sec(kky) &&
GPS.y.msec(kk)==GPS.y.rawdata.msec(kky)
        GPS.y.latitude(kk)=GPS.y.rawdata.latitude(kky);
        GPS.y.longitude(kk)=GPS.y.rawdata.longitude(kky);
        GPS.y.altitude(kk)=GPS.y.rawdata.altitude(kky);
        GPS.y.heading(kk)=GPS.y.rawdata.heading(kky);
        GPS.y.speed(kk)=GPS.y.rawdata.speed(kky);
        kky=kky+1;
    else
        GPS.y.latitude(kk)=GPS.y.latitude(kk-1);
        GPS.y.longtiude(kk)=GPS.y.longitude(kk-1);
        GPS.y.altitude(kk)=GPS.y.altitude(kk-1);
        GPS.y.heading(kk)=GPS.y.heading(kk-1);
        GPS.y.speed(kk)=GPS.y.speed(kk-1);
    end
end
dt=0.1; % [sec]
%% Create Helicopter Path
GPS.helipath.Vx.raw=GPS.h.speed.*sind(GPS.h.heading)*6080.4/3600; % [ft/s]
GPS.helipath.Vy.raw=GPS.h.speed.*cosd(GPS.h.heading)*6080.4/3600; % [ft/s]
for k=1:length(GPS.helipath.Vx.raw)-1

GPS.helipath.Vx.avg(k)=.5*(GPS.helipath.Vx.raw(k)+GPS.helipath.Vx.raw(k+1));
% [ft/s]

GPS.helipath.Vy.avg(k)=.5*(GPS.helipath.Vy.raw(k)+GPS.helipath.Vy.raw(k+1));
% [ft/s]
end
GPS.helipath.posx(1)=0;
GPS.helipath.posy(1)=0;
for k=2:length(GPS.helipath.Vx.raw)
    GPS.helipath.posx(k)=GPS.helipath.posx(k-1)+GPS.helipath.Vx.avg(k-1)*dt;
    GPS.helipath.posy(k)=GPS.helipath.posy(k-1)+GPS.helipath.Vy.avg(k-1)*dt;
end
%% Create YP Path
GPS.YPpath.Vx.raw=GPS.y.speed.*sind(GPS.y.heading)*6080.4/3600; % [ft/s]
GPS.YPpath.Vy.raw=GPS.y.speed.*cosd(GPS.y.heading)*6080.4/3600; % [ft/s]
for k=1:length(GPS.YPpath.Vx.raw)-1
    GPS.YPpath.Vx.avg(k)=.5*(GPS.YPpath.Vx.raw(k)+GPS.YPpath.Vx.raw(k+1)); %
[ft/s]
    GPS.YPpath.Vy.avg(k)=.5*(GPS.YPpath.Vy.raw(k)+GPS.YPpath.Vy.raw(k+1)); %
[ft/s]
end
GPS.YPpath.posx(1)=0;
GPS.YPpath.posy(1)=0;
```

```
for k=2:length(GPS.YPpath.Vx.raw)
    GPS.YPpath.posx(k)=GPS.YPpath.posx(k-1)+GPS.YPpath.Vx.avg(k-1)*dt;
    GPS.YPpath.posy(k)=GPS.YPpath.posy(k-1)+GPS.YPpath.Vy.avg(k-1)*dt;
end
%% Convert to Relative Flight Deck Location
GPS.relative.distance=zeros(1,length(GPS.YPpath.posx));
GPS.relative.theta=zeros(1,length(GPS.YPpath.posx));
for k=2:length(GPS.YPpath.posx) % combination of pythagorean theorem and law
of cosines
    a=sqrt((GPS.helipath.posx(k)-GPS.YPpath.posx(k))^2+(GPS.helipath.posy(k)-
GPS.YPpath.posy(k))^2);
    b=sqrt((GPS.YPpath.posx(k)-GPS.YPpath.posx(k-1))^2+(GPS.YPpath.posy(k)-
GPS.YPpath.posy(k-1))^2);
    c=sqrt((GPS.YPpath.posx(k-1)-GPS.helipath.posx(k))^2+(GPS.YPpath.posy(k-
1)-GPS.helipath.posy(k))^2);
    theta=acosd((a^2+b^2-c^2)/(2*a*b));
    GPS.relative.distance(k)=a; % [ft]
    GPS.relative.theta(k)=real(theta); % [deg] straight aft is zero degrees
    if GPS.h.heading(k)>GPS.y.heading(k)
        GPS.relative.theta(k)=-GPS.relative.theta(k);
    end
end
GPS.relative.posx=GPS.relative.distance.*sind(GPS.relative.theta);
GPS.relative.posy=GPS.relative.distance.*cosd(GPS.relative.theta)+10;
%% Plots
H2=figure('Name','Helicopter Flight Path Relative to YP');
hold on
axis equal
line([-11.0,11.0],[0,0],'color','k','linewidth',2)
line([-10.5,10.5],[20,20],'color','k','linewidth',2)
line([-11.0,-10.5],[0,20],'color','k','linewidth',2)
line([11.0,10.5],[0,20],'color','k','linewidth',2)
plot(GPS.relative.posx,GPS.relative.posy,'b-','linewidth',1)
xlabel('Lateral Distance (ft)')
ylabel('Longitudinal Distance (ft)')
grid on

H3=figure('Name','Helicopter Flight Path on CFD Picture');
hold on
CFDimg=imread('beta15hangarCFD.jpg');
image([-25,25],[0,55],CFDimg)
plot(GPS.relative.posx,GPS.relative.posy,'b-','linewidth',2)
xlabel('Lateral Distance (ft)')
ylabel('Longitudinal Distance (ft)')
axis([-25 25 0 55])
```